Anonymous Authenticated Car-to-X Communication

Vom Fachbereich Informatik
der Technischen Universität Darmstadt
genehmigte

Dissertation

zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.)
von

Carsten Gerhard Büttner, M.Sc.

geboren in Aschaffenburg, Deutschland

Referenten der Arbeit:	Prof. Dr.-Ing. Sorin A. Huss Technische Universität Darmstadt
	Prof. Dr.-Ing. Delphine Reinhardt Rheinische Friedrich-Wilhelms-Universität Bonn
	Prof. Dr. rer. nat. Max Mühlhäuser Technische Universität Darmstadt
Tag der Einreichung:	22. September 2016
Tag der mündlichen Prüfung:	04. November 2016

Darmstädter Dissertation 2016
D 17

Erklärung zur Dissertation

Hiermit versichere ich, die vorliegende Dissertation selbständig nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 22. September 2016 Carsten Gerhard Büttner

Acknowledgment

I would like to thank Prof. Dr.-Ing. Sorin A. Huss for his guidance and support. Especially for his effort in improving my scientific writing in English. I also thank Prof. Dr.-Ing. Delphine Reinhardt to be available as the second and Prof. Dr. rer. nat. Max Mühlhäuser to be available as third assessor of this thesis. Additionally, I thank Prof. Dr. phil. Iryna Gurevych, Prof. Dr.-Ing. Andreas Koch, and Prof. Dr. rer. nat. Andy Schürr to be part of my committee.

Thanks to my Ph.D. colleagues at the Adam Opel AG, namely Lena Rittger, Jens Ferdinand, Thomas Streubel, Rami Zarife, Jonas Firl, Boliang Yi, Bernhard Wandtner, and Maximilian Harr. Moreover, I would like to thank my colleagues at the Integrated Circuits and Systems Lab at TU Darmstadt, Tolga Arul, Attila Jaeger, Alexander Biedermann, Zheng Lu, and Qizhi Tian as well as my colleague Marco Grimm at CASED.

I further thank my students for their contributions - namely Friederike Bartels, Marc Schiller, Christoph Brenner, Jakob Laenge, Johannes Wagener, Yasser Aziza, Josef Daher, and Matthias Braun.

Many thanks to Harald Berninger, who supervised my work at the Adam Opel AG and reviewed parts of this thesis. Furthermore, I would like to thank Bernd Büchs for his helpful advices in the practical implementations and our endless discussions about privacy and security not limited to Car-to-X Communication.

Special thanks to Tobias Rückelt for all the discussions about the topic and reviewing this thesis and many papers. I further thank Frank Englert, Heike Perko, Sabine Schäfer, Norman Göttert, and Burkhard Hauck for reviewing parts of this thesis.

I also thank my friends who encouraged me on my way. Additionally, I would like to thank my family for their support during the whole time.

Carsten

Abstract

Two current trends in the automotive industry are the increasing number of connected vehicles and automated driving. The former enables the use of different applications within the vehicle. These applications might be restricted to vehicles with certain features such as manufacturer or model.

To enable automated driving, the vehicle needs information about the road ahead. This information might be provided by an application. In order to keep the street information up to date connected vehicles share their sensor data. This data is then aggregated on a central server. Furthermore, it has a restricted spatial and temporal validity. Therefore, the vehicles also need to provide the corresponding time and position information.

When reporting position data, it is possible, for example, to generate movement profiles or to identify sensitive locations. Hence, it is critical which information different applications reveal about the corresponding vehicles.

Therefore, in this thesis we propose four different schemes which restrict the information applications can obtain from vehicles. The first scheme addresses the problem how a vehicle can authenticate itself privacy preserving based on attributes at an application without revealing all its attributes. The second scheme provides a solution for the question how two vehicles can authenticate each other for an application and exchange confidential data without disclosing their identity. The third scheme obfuscates the identity of a vehicle while sharing sensor data with a central server. The fourth scheme is related to the question how data can be distributed by a central server to all vehicles equipped with a particular application and located within a certain area without tracking the vehicles and knowing their subscribed applications. In addition, we outline how these schemes can be combined.

We demonstrate that each scheme is practical by presenting prototype implementations. Additionally, we simulate the second and third scheme in order to assess the impact on the vehicles privacy.

Zusammenfassung

Zwei aktuelle Trends in der Automobilindustrie sind einerseits die zunehmende Vernetzung der Fahrzeuge und andererseits das automatisierte Fahren. Durch ersteres wird es ermöglicht, verschiedene Anwendungen im Fahrzeug zu nutzen. Diese Anwendungen können auf Fahrzeuge mit bestimmten Eigenschaften wie Hersteller oder Modell beschränkt sein.

Das automatisierte Fahren benötigt Informationen über die voraus liegende Strecke, welche über eine Anwendung bereitgestellt werden. Um die Straßeninformationen auf aktuellem Stand zu halten, stellen die vernetzten Fahrzeuge Sensordaten bereit, die schließlich auf einem zentralen Server gesammelt werden. Da diese Daten eine begrenzte räumliche und zeitliche Gültigkeit haben muss das Fahrzeug auch die zugehörigen Zeit- und Positionsinformationen bereitstellen

Durch die Weitergabe von Positionsdaten können jedoch beispielsweise Bewegungsprofile erstellt oder sensible Orte identifiziert werden. Daher ist es ein kritischer Aspekt, welche Informationen die verschiedenen Anwendungen über die Fahrzeuge preisgeben.

Deshalb werden in dieser Arbeit vier Verfahren vorgestellt, um die weitergegebenen Informationen gezielt einzuschränken: Das erste Verfahren beschäftigt sich mit der Frage, wie sich ein Fahrzeug bei gleichzeitiger Wahrung der Privatsphäre, basierend auf Attributen für eine Anwendung, authentifizieren kann, ohne alle seine Eigenschaften zu offenbaren. Das zweite Verfahren erlaubt es, dass sich zwei Fahrzeuge gegenseitig für eine Anwendung authentifizieren und vertrauliche Daten austauschen, ohne ihre Identität zu offenbaren. Das dritte Verfahren verschleiert die Identität eines Fahrzeuges, welches Sensordaten an einen zentralen Server sendet. Das vierte Verfahren ermöglicht es einem Server, Informationen an alle Fahrzeuge mit einer speziellen Anwendung in einem bestimmten Gebiet zu senden, ohne den Standort der Fahrzeuge oder die installierten Anwendungen zu kennen. Zusätzlich zeigen wir, wie diese Verfahren miteinander kombiniert werden können.

Mithilfe prototypischer Implementierungen demonstrieren wir, dass jedes dieser Verfahren praxistauglich ist. Für das zweite und dritte Verfahren führen wir zur Beurteilung der Auswirkungen auf die Privatsphäre zusätzlich Simulationen durch.

Contents

Acronyms

AA	Authorization Authority
AC	Anonymous Credential
AES	Advanced Encryption Standard
AGfIA	Anonymous Geocast scheme for ITS Applications
AID	Application ID
AT	Authentication Ticket
AU	Application Unit
BM-SC	Broadcast Multicast Service Center
BTP	Basic Transport Protocol
C2C	Car-to-Car
C2C-CC	Car-2-Car Communication Consortium
C2I	Car-to-Infrastructure
C2X	Car-to-X
CA	Certificate Authority
CAM	Cooperative Awareness Message
CAMP	Crash Avoidance Metrics Partnership
CCU	Communication and Control Unit
CONVERGE	COmmunication Network VEhicle Road Global Extension
CRL	Certificate Revocation List
DAA	Direct Anonymous Attestation
DENM	Decentralized Environmental Notification Message
DSRC	Dedicated Short Range Communication
EA	Enrolment Authority
ECIES	Elliptic Curve Integrated Encryption Scheme
ECC	Elliptic Curve Cryptography
EC	Enrolment Credential
ECDSA	Elliptic Curve Digital Signature Algorithm
eNodeB	evolved Node B
EPID	Enhanced Privacy ID

ETSI	European Telecommunications Standards Institute
GAC	Geographically-Scoped Anycast
GBC	Geographically-Scoped Broadcast
GBGS	Grid Based Geocasting Scheme
GMS	Geo Messaging Server
GN6	GeoNetworking-IPv6
GPS	Global Positioning System
GUC	Geographically-Scoped Unicast
GUI	Graphical User Interface
HSM	Hardware Security Module
ICS	ITS Central Station
ICS EA	ICS Enrolment Authority
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
IRS	ITS Roadside Station
IRS CS	IRS Central Station
ITS	Intelligent Transportation Systems
IVS	ITS Vehicle Station
IVS AA	IVS Authorization Authority
IVS EA	IVS Enrolment Authority
LTE	Long Term Evolution
MAC	Message Authentication Code
MBMS	Multimedia Broadcast Multicast Service
MBMS-GW	MBMS-Gateway
MCE	Multicell Coordination Entity
MEC	Mobile Edge Computing
MME	Mobility Management Entity
MN CS	Mobile Network Central Station
MNO	Mobile Network Operator
MTU	Maximum Transmission Unit
NIST	National Institute of Standards and Technology
OEM	Original Equipment Manufacturer
OSI	Open Systems Interconnection
OSM	OpenStreetMap
PK	Public Key
PKI	Public Key Infrastructure
PUF	Physical Unclonable Function
RSA	Rivest, Shamir, and Adleman
SAM	Service Announcement Message
SD	Service Directory

SSP	Service Specific Permission
TLS	Transport Layer Security
TTP	Trusted Third Party
UDP/IP	User Datagram Protocol/Internet Protocol
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
VANET	Vehicular Ad-hoc NETwork
VM	Virtual Machine
WAVE	Wireless Access in Vehicular Environments
WLAN	Wireless Local Area Network

1 Introduction

'The auto industry is poised for more change in the next five to ten years than it's seen in the past 50'

Marry Barra, 2015

Marry Barra gave this statement as CEO of General Motors at the 2015 Code Conference. Among others, connected vehicles and automated driving will drive this change. However, both introduce privacy risks for the driver which have to be solved. In this thesis we propose solutions to protect the privacy. Before outlining these risks, we will first describe the terms connected vehicles and automated driving.

A connected vehicle is able to communicate with entities located in the Internet. First, vehicles provided simple services like driving directions or emergency assistance via mobile networks [GM 01]. Nowadays, vehicles provide all kinds of services like real time traffic information, advanced diagnostics, news, or Internet access [BMW], [GM]. Some systems allow drivers to even install applications form a marketplace [GM14].

These marketplaces allow users to install convenience, entertainment, or even safety applications. Convenience applications might allow the passengers to, e.g., reserve parking lots, get periodic diagnostic reports, or pay automated at tolling stations [CON15b] [GM] [ETS09b]. Entertainment applications enable music streaming, web browsing, or even video on demand services. Applications to enhance the safety might provide weather hazard warnings, traffic jam information, or wrong way driver warnings [ETS09b] [JH11].

When a vehicle takes partly or fully control of driving, it is called automated driving [VDA15]. Different levels of automated driving exist, from assisted to full automation. Some development vehicles with a high level of automation already exist [ZBS+14], [Urm14]. One important source of information for automated driving vehicles are up-to-date high-precision maps. They contain detailed information about the curves, lanes, traffic signs, road works, or traffic lights. Therefore, all these vehicles will be equipped with an application, in order to get this information from a

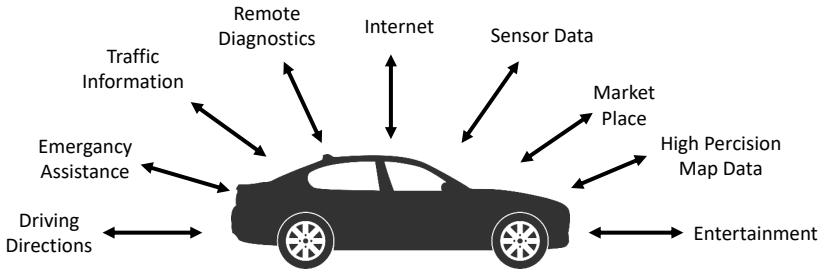


Figure 1.1.: Current and future example applications of connected vehicles

central server. Some provider for detailed up to date map data already exist [HER16].

However, the map data needs to be kept up to date. Old data might increase the number of unclear situations a vehicle encounters. This increases the risk of accidents. To keep the data up to date, built in sensors like camera or wheel sensor of the vehicle can be exploited. A camera can for example detect changed traffic signs or lane marks. A wheel sensor can detect changes in the surface. Detected changes can then be send to a central server. The server can aggregate the data received from different vehicles. Afterwards, it might send the changed information to all vehicles located in the relevant area [Dub15].

Finally, future vehicles will support users in basic driving tasks by automation but also in strategic driving efficiency and convenience areas, e. g. by parking lot reservation services or entertainment. This way, automatic driving in combination with connected vehicle will change future driver experience. Different current and future applications used by connected vehicles are illustrated in Figure 1.1.

In the sequel, we first motivate why the solutions to protect the privacy of vehicles and drivers proposed in this thesis are necessary. Afterwards we outline the requirements of the vehicles and central servers on such a system. Finally, we conclude the introduction with the outline of the thesis and our contributions.

1.1. Motivation

Automated driving and connected vehicle applications provide great potentials to improve traffic safety and driving convenience. However, they also introduce privacy risks on sensible information about the vehicle and driver. These risks originate both from direct communication between vehicles and from communication to central servers.

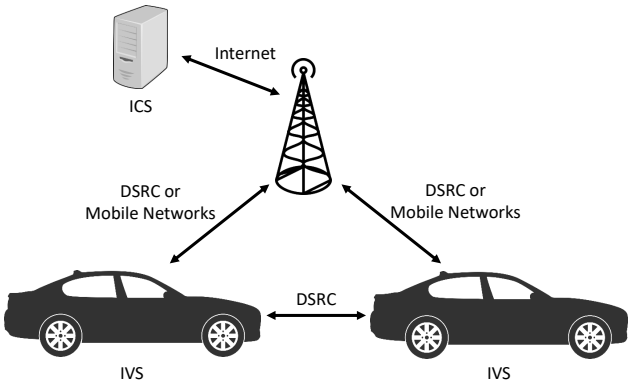


Figure 1.2.: Communication system

To enable a direct communication between vehicles a special Wireless Local Area Network (WLAN), called Dedicated Short Range Communication (DSRC) can be applied. For the communication with a central server mobile networks or DSRC via infrastructure might be exploited. A vehicle equipped with multiple communication technologies might always select the most suitable one if multiple are available. Possible factors which influence the selection are the provided Quality of Service and transmission costs [WK13]. If vehicles communicate with other vehicles, the infrastructure, and central servers it is called Car-to-X (C2X) communication. Systems that allow this kind of communication are part of Intelligent Transportation Systems (ITS). Each vehicle participating in the system is called ITS Vehicle Station (IVS), the server of an application ITS Central Station (ICS). This communication system is illustrated in Figure 1.2.

Different areas of applications exist: convenience, entertainment, and safety. The exchange of data for safety applications between IVSs via DSRC has already been well studied [ETS09b] [sim13] [DRI14]. In contrast, we focus in this thesis on the distribution of non-safety data between IVSs and the general communication with ICSs.

Some applications might restrict their usage to IVSs with certain attributes like a specific brand or charge a fee for service usage. In order to support these restrictions, the application or the marketplace needs to validate the attributes of the IVSs. These attributes might include sensible data like the brand, model, sensors or even the production date of the IVS. A naive solution to validate the attributes would be that the application or marketplace is aware of all attributes of an IVS.

An application might be interested in sensor data of its subscribed IVSs to update

the map information it provides. Later on it aggregates this data and distributes the results to other IVSs in the relevant geographic area. The data to be reported, like road works or changed traffic signs, has to be recognized by the built in sensors of the IVS first. Afterwards, the IVS authenticates itself at the ICS and sends the reported data to it. Each detected change in the provided street data is spatial and temporal limited. A weather hazard or the start of road works are for example restricted to a fixed point. Furthermore, these events are only relevant for a limited period of time. Danger from an icy road in the morning might for example disappear if the temperature increases during the day. Therefore, an IVS needs to send the location and time of the detected change together with the changed information to the ICS. Furthermore, the ICS needs to ensure the data is sent from a valid IVS. Therefore, the data needs also to be authenticated.

After the ICS received the changed data, it distributes information updates to all IVSs located in the relevant geographic area. A common means to identify the right recipients for message dissemination is to keep track of the location of all IVSs [JRX11] [FWZ12]. Each time data has to be distributed, the sender looks up all IVSs located in this area and transmits the data to them.

To get up to date data IVSs might also exchange data directly with each other. This communication needs also to be authenticated. Otherwise an IVS will not trust the data. This data may influence automated driving maneuvers and therefore the safety of the driver and his environment. The authentication might be done by applying certificates and digital signatures. Furthermore, some data might be confidential and require additional protection.

If an application or marketplace is *aware of all attributes* of an IVS this information might be exploited in order to harm the driver. An application provider might for example charge a higher price for the same application to drivers of more expensive vehicles [HSL+14]. An attacker could also exploit this information to identify certain vehicle models he is aiming to steal. The transmitted sensor data furthermore *contains sensible information*. Without a suitable protection this data might be accessed and misused by third parties. Since the data is only valid for a small area, it is tagged with a geographic location. Additionally, the data includes the collection time. Even if the data is not tagged with a time, an ICS could record the point in time when it received the data. Subsequently, the ICS is able to record the visited locations of the reporting IVS. In addition, the distribution of data in certain geographic areas may also *leak the position* of the IVSs.

A vehicle is a personal object, which is taken to *sensible locations* like home, work or hospitals. If position data is sent to or collected by an ICS, it is possible to create movement profiles of the driver or to identify often visited locations [Kru07]. If the location samples are furthermore tagged with time, other sensible information like speeding can be detected [Lae15].

An attacker might also misuse the identity included in the authenticated messages exchanged directly between IVSs to track an IVS.

Hence, it is critical, which information IVSs reveal. Attributes as well as their identity or location and time information might be exploited in order to harm the driver. However, the attributes of an IVS or the reported position data do not necessary need to be linked to the identity of the IVS. More sophisticated mechanisms as the outlined above might be applied in order to prevent the described risks.

We propose different schemes to limit the information third parties can get from an IVS in this thesis. These schemes protect the privacy of the IVSs and ensure the authenticity of the transmitted data at the same time. We developed a system, where an IVS can authenticate for applications based on attributes by only revealing the necessary ones. Furthermore, we propose a mechanism to hide the originating IVS of reported sensor data. We also present a scheme which allows the dissemination of data to all IVSs running a specific application and located in a certain geographic region without tracking the IVS. In addition, we propose an anonymous authenticated key agreement protocol which allows two IVSs to authenticate each other and exchange confidential data without revealing their identity.

1.2. Requirements and Challenges

In the sequel we outline the most important requirements of the ICS and IVS in the detailed communication system.

For an ICS it is important that all data received from IVSs is authenticated. If the data is not authenticated, it would not be possible to determine how trustworthy it is. If sensor data is reported from an IVS the ICS needs to know its exact position in order to process it. The communication needs also be secured in a way that no third entity is able to extract the transmitted data. Another requirement of the ICS is the reception of up to date data. Therefore, the end-to-end delay for the communication should be minimized. If the transmission delay is lower, the ICS is able to distribute the data fast enough to other IVSs in the relevant geographic area. To disseminate data to all subscribed IVSs present in the area a suitable mechanism is necessary. In order to encourage a possible ICS to offer an application it is necessary to provide a way to bill the subscribed IVSs for their application usage. Some ICSs might further aim in offering their application only to IVSs with certain properties. Possible properties are the Original Equipment Manufacturer (OEM), brand, and sensor availability. Furthermore, information like the IVS manufacturing date are of interest to, e. g., give special offers to new IVSs. Therefore, the system must offer some kind of service to enable an ICS to check the attributes of the IVSs. To get more subscribers, the ICS aims in a simple, efficient mechanism to advertise its applications. Further-

more, the system should be able to scale for a huge amount of IVSs as receivers of messages. ITS applications can be of interest for several millions of IVSs.

One very important requirement of the IVS is its privacy. Different privacy types exist [FWF13]. Several of them are relevant in the outlined system: identity, communication, location and space, and association. The IVS aims in hiding its identity to every entity, which does not require it. An ICS which offers for example a weather application does not need to know the identity of the subscribed IVSs. The identity also includes its characteristics like sensors and features. Whenever applications require certain attributes of the IVSs, only the necessary information should be provided to the ICS. The communication shall also ensure the privacy of the IVS. An attacker shall not be able to get the payload transmitted between the IVS and an ICS. Furthermore, the privacy of location and space is important for the IVS. An ICS should not be able to track an IVS by its reported data or require periodic location samples. All functions should be designed in a way that this is not possible to track an IVS. The last relevant privacy aspect of the IVS to the outlined system is the privacy of association. It shall not be possible for any entity to obtain information of the applications which an IVS is subscribed to. Because of cost pressure current IVSs feature only a limited computational power. Therefore, only the necessary operations should be executed on the IVS. All other operations should be offloaded to static entities with more computational power, if possible. In order to identify new applications of interest, an IVS should be able to search for new applications. Another requirement of the IVS is the possibility to ensure that it is really communicating with the ICS. Otherwise an attacker might send false information to IVSs or an IVS might send confidential data to an attacker.

The ICS requires up to date authenticated data only relevant in a small area. Furthermore, mechanisms to restrict the subscribers by certain attributes and disseminate data to IVSs located in a certain area are required. On the other hand, the IVS requires privacy, which includes hiding of the own identity, location and space, and associations. Therefore, the requirements of the ICS and IVS are conflicting. This introduces different challenges to the communication system:

C.1 How can the ICS ensure that an IVS features certain attributes, while an IVS aims in hiding its attributes?

C.2 How can the data sent from IVSs be authenticated, and their identities be protected at the same time?

C.3 How can the visited locations of an IVS be protected, while reporting sensor data tagged with location information to an ICS?

C.4 How can data be distributed to IVSs located in a specific geographic area, without tracking their movements?

Table 1.1.: Aligned and conflicting requirements of ICS and IVS

ICS	IVS	Challenge
Confidentiality	Privacy of communication	-
Application advertising	Search for applications	-
Restriction by attributes	Privacy of the identity	C.1
Authenticated data	Privacy of the identity	C.2
Location tagged data	Privacy of location and space	C.3
Geographical data dissemination	Privacy of location and space	C.4

However, some of the requirements are also aligned. The ICS requires for example a mechanism to offer services and the IVS a mechanism to search for services. Both entities also aim in a secured communication. A summary of the aligned and conflicting requirements is given in Table 1.1. In this thesis we provide solutions to the outlined challenges without violating the other requirements.

1.3. Structure and Contributions

Basic knowledge of different security algorithms, protocols, and mechanisms is provided in Chapter 2. It furthermore gives an introduction to C2X communication and the Multimedia Broadcast Multicast Service (MBMS).

Chapter 3 reviews the state of the art for the proposed schemes.

The following four chapters propose different solutions to the outlined challenges. Each scheme protects a different communication scenario: the privacy preserving authentication at an application, the secure authenticated communication between IVSs, the reporting of data without leaking the originating IVS, and the anonymous distribution of data from an ICS to all registered IVSs located in a geographic area.

In Chapter 4 we propose a system which allows an IVS to authenticate at an ICS anonymously based on attributes. This system allows an IVS to prove the possession of attributes while protecting its identity (Challenge C.1). Furthermore, it allows an IVS to authenticate at an ICS while hiding its real identity (Challenge C.2). The system applies Anonymous Credentials (ACs) to obtain Authentication Tickets (ATs) from a Public Key Infrastructure (PKI) for a specific application. ACs allow to restrict the IVSs exploiting an application by attributes. An ICS defines the attributes which an IVS must possess in order to obtain ATs for the application. When an IVS wants to get ATs for this application, it proves them by its AC, without revealing the other attributes or its identity. These ATs might then be exploited to authenticate for this application. Furthermore, no entity is able to figure out the identity of the IVS employing the ATs. These tickets are issued in a way that it is not possible to

exploit them at any other application than the intended one. All other applications will reject the ATs. Furthermore, the system supports different methods to bill the usage of applications. We evaluate its suitability by a prototype implementation. In summary, the scheme protects the privacy of the IVS and enables at the same time authentication and access restriction by attributes.

When confidential data is exchanged between two IVSs both have to authenticate each other for the application first. However, they aim at hiding their identity at the same time (Challenge C.2). We present in Chapter 5 a novel anonymous authenticated key agreement protocol to enable a confidential exchange of data between two IVSs exploiting the same application while hiding their identity. It is based on the Elliptic Curve Integrated Encryption Scheme (ECIES) and ring signatures. ECIES is exploited to agree on a symmetric encryption key between the IVSs to exchange confidential application data. Ring signatures are applied to hide the identity during authentication. To authenticate each other for the application the IVSs exploit the ATs introduced in the previous chapter. We show by simulation that ring signatures prevent an attacker from identifying the participating IVSs. Furthermore, we created a prototype implementation to evaluate the protocol on real vehicles. Therefore, this chapter solves the challenge of a privacy preserving confidential data exchange between two IVSs.

When an IVS reports data to an ICS it aims at protecting its privacy of location and space. The ICS might however require sensor data tagged with location and time (Challenge C.3). In Chapter 6 we exploit the protocol outlined in Chapter 5 to exchange recorded sensor data between IVSs prior to sending them to an ICS. By exchanging the sensor data, it is no longer possible for the ICS to determine the origin IVS of the data. Therefore, it is no longer possible to assign the data to a specific IVS and, e. g., track an IVS or create movement profiles. We propose two different strategies on how to exchange the data between IVSs before sending it to the ICS. Furthermore, we created a simulation scenario to assess the impact on the privacy. In addition, we successfully run one strategy on real vehicles. This scheme enables an IVS to report sensor data tagged with time and position to an ICS while protecting its privacy of location and space.

ICSs aim at distributing data to all registered IVSs located in a certain geographic area. The IVSs however do not want to reveal their positions (Challenge C.4). In Chapter 7 we outline an anonymous geocast scheme which solves this challenge. This scheme allows an ICS to send messages to all IVSs running a particular application and located in a certain geographic region. This is done without the knowledge of the IVSs present in the target area. Furthermore, this scheme does not track the applications installed by an IVS. Moreover, it supports different communication technologies like Long Term Evolution (LTE) and DSRC for distribution. We also implemented it on real devices to show its suitability on real world scenarios. There-

fore, it enables an ICS to distribute data to all registered IVSs located in a specific geographic area without hurting their privacy.

Finally, Chapter 8 concludes this thesis with a summary and identifies areas of future research.

All developed concepts have already been published by the author of this thesis. The relevant publications for each scheme are cited in the respective chapter. In addition to the publications this thesis outlines how the different schemes relate to each other and can be combined. Furthermore, enhancements, additional evaluations, and more detailed descriptions are presented in this thesis. Text passages taken out of cited own publications are original work of the author of this thesis.

2 | Foundations

In this chapter we first define the term privacy and describe the privacy aspects we aim to protect in this thesis. Afterwards, we briefly describe the security mechanism employed in this thesis. Then, for the application of these mechanisms we give an introduction in C2X communication. Finally, we describe how multicast messages can be distributed in LTE.

2.1. Privacy

Different definitions of privacy exist [WB90][Wes67][FWF13]. In general privacy can be seen as the right to determine which personal information are shared with which entities. Companies are interested in this personal information to process and get a benefit out of it. However, the information belongs to the user. Because this information can be exploited to harm the user, he is not willing to provide it. Therefore, the privacy has to be protected. In order to properly protect the privacy of an individual it hast to be determined how the privacy is threatened first.

The authors of [FWF13] categorize privacy in seven different types: *privacy of the person*, *privacy of behaviour and action*, *privacy of communication*, *privacy of data and image*, *privacy of thoughts and feelings*, *privacy of location and space*, and *privacy of association* (including group privacy). They are defined by the authors as follows:

> *The privacy of the person covers the protection of body functions and characteristics like genetic codes and biometrics. The protection of personal sensitive topics like political activities, religious practices, or sexual preferences and habits are encompassed by the privacy of behavior and action. Privacy of communication intents to prevent communication interception, the use of bugs, directional microphones, or the recording and access to transmitted data. Privacy of data and image aims in giving people control over their data and make it not automatically available to other individuals and organizations. The right of people not to share and reveal*

> *their thoughts and feelings is covered by the privacy of thoughts and feelings. The privacy of location encompasses the right to move in public space without being tracked, identified, or monitored. Individuals right to associate with anyone without being monitored is covered by the privacy of association (including group privacy).*

The relevant types of privacy for the outlined system are the *privacy of the person*, *privacy of communication*, *privacy of location and space*, and *privacy of association*. In the sequel we outline, why they are relevant for the outlined system.

Even if an IVS is not a person, the *privacy of the person* can still be applied. Most of the time an IVS is operated by the same person. Moreover, the IVS possesses attributes, which need to be protected from third parties. In addition to the definition in [FWF13] we also consider the privacy of the identity as part of the privacy of the person. Therefore, an IVS also aims in hiding its identity from third parties which do not require to know it.

In the outlined system sensitive, confidential, and even private data might be transmitted between the different entities. Therefore, the *privacy of communication* has to be protected. A third party with access to this data might exploit it in order to harm the different entities.

If an entity gets location samples of IVSs, it can create movement profiles. Furthermore, IVSs are taken by the driver to sensitive locations like home, work place, or medicals. From this information it is possible to derive the identity of the driver. The obtained information might then be exploited to harm the driver. Therefore, the *privacy of location and space* is important in the outlined system.

The driver of an IVS can install applications for additional services like parking lot reservation, toll payment, or music streaming. This can be seen as the association to the group of users of this application. A third entity might exploit this information in order to discriminate the user by, e.g., charge higher prices for other services or excluding him from offers. Therefore, the privacy of the driver can be harmed in the outlined system, if the *privacy of association* is not protected.

For each considered communication scenario in this thesis we outline the affected types of privacy. Furthermore, we also describe how the proposed schemes protect each affected type of privacy.

k-Anonymity

It is difficult to decide based on given data whether an entity is anonymous or not. To overcome this issue Sweeney defined in [Swe02] K-ANONYMITY as follows:

> *k-anonymity is a metric for the anonymity of an entity, where k denotes the number of entities it is indistinguishable from.*

We exploit this metric in order to measure the anonymity of the IVSs in our developed schemes.

2.2. Security and Cryptography

Whenever a communication needs to be secured, cryptographic mechanisms are necessary. Two different types of cryptography exist, SYMMETRIC and ASYMMETRIC [Buc04]. They can be defined as follows:

> *When applying symmetric cryptography, the same key is applied to encrypt and decrypt the data. Therefore, the key needs to be kept private. Asymmetric cryptography uses different keys for encrypting and decrypting data, called private and public key. They are mathematically related to each other. The public key is applied to encrypt the data. This key is public available. The private key is exploited to decrypt cypher text encrypted with the corresponding public key. This key is kept private. Everyone with access to this key can decrypt the data.*

An attacker with access to the public key is not able to derive the corresponding private key.

Symmetric vs. Asymmetric Crpytography

The advantages of symmetric over asymmetric cryptography is the more efficient calculation. When applying asymmetric cryptography complex and resource expensive operations are necessary. Therefore, the calculation takes longer or more powerful hardware is necessary. However, the disadvantage of symmetric cryptography is the key exchange. The applied key has to be exchanged over a secure channel. Everyone eavesdropping the key is able to decrypt the exchanged data. When asymmetric cryptography is applied no secure channel is necessary. There the public key is no sensitive information. Another disadvantage of symmetric cryptography is the key management. Because the same key is applied for encryption and decryption, a unique key is necessary for each communication partner. If a system consists of n participants, $n(n-1)/2$ keys are necessary when everyone wants to communication with everyone else. When asymmetric cryptography is applied, only the own private and the public key of each communication partner is necessary.

Therefore, asymmetric cryptography is normally applied to exchange a symmetric encryption key between two entities. The exchanged data is then encrypted with this symmetric key. In this way the key exchange problem is solved by asymmetric cryptography and the communication partners can benefit from the lower computational overhead of symmetric cryptography. In this thesis Advanced Encryption Standard (AES) [NIS] is always applied as the symmetric encryption algorithm. Asym-

metric algorithms based on Elliptic Curve Cryptography (ECC) feature a shorter key length in comparison to other cryptographic problems while maintaining the same level of security. In order to further reduce the signature size, elliptic curve point compression [VMA00] can be applied. Because the channel capacity in C2X communication is very limited ECC is exploited when possible. Therefore, algorithms based on ECC are preferred in this thesis.

Digital Signatures

Besides confidentiality, which is ensured by encryption, it is also desirable to authenticate a communication and ensure its integrity. For this, DIGITAL SIGNATURES can be used [Buc04]. They can be defined as follows:

> *Digital signatures are based on asymmetric cryptography. In comparison to encryption the private key of an entity is exploited by the signer in order to create the digital signature. A verifier can then validate the signature by applying the public key. After validating the signature, the verifier is convinced of the authenticity that the message is from the sender, and integrity that the message was not altered, of the message.*

A common algorithm for digital signatures based on ECC is Elliptic Curve Digital Signature Algorithm (ECDSA) [JMV01]. When applying symmetric cryptography, a Message Authentication Code (MAC) can be created in order to ensure the integrity of the transmitted data.

Digital Certificates

When applying asymmetric cryptography, it is necessary to ensure that a public key belongs to a certain entity. The identity of a public key owner cannot be derived from the key itself. This can be however achieved by DIGITAL CERTIFICATES [Buc04]. They can be defined as follows:

> *In order to assigning a public key to an entity, the public key is signed together with identity information, like an email address, from a trusted third party. The information together with the signature is called digital certificate.*

Then everyone who trusts the third party can verify that the public key belongs to the claimed entity. Besides the identity a digital certificate can also contain other information about the entity.

Public Key Infrastructure

It is difficult to decide if a third party issuing a certificate is trustworthy. In principle everyone can issue a digital certificate containing a public key with any identity. In

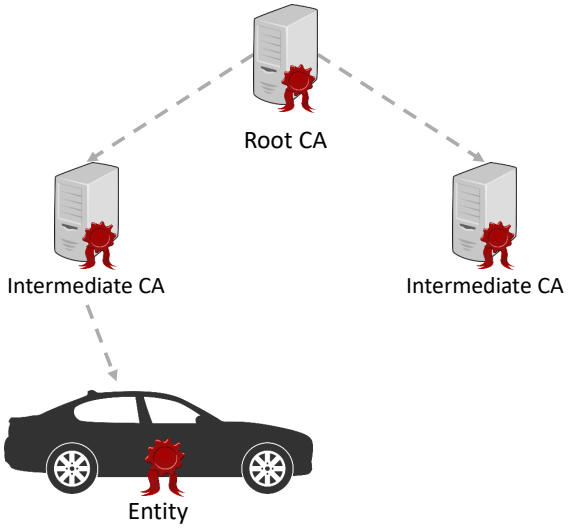


Figure 2.1.: Structure of a PKI

order to better decide whether an issuer of a certificate is trustworthy PKIs can be used [Buc04]. They can be defined as follows:

> *PKIs are trustworthy third parties issuing certificates. They verify if a public key belongs to the claimed identity.*

The basic structure of an PKI is depicted in Figure 2.1. They consist of different entities where each is called a Certificate Authority (CA). One of them is called Root CA. This is the so called trust anchor. Generally, an entity trusts this Root CA. Normally, this CA issues digital certificates for so called Intermediate CAs. These Intermediate CAs issue certificates either to other Intermediate CAs or to be certified entities. Since a verifier of a digital certificate trusts the Root CA, he also trusts the Intermediate CAs. When the Intermediate CAs are trusted, their issued certificates are also trusted for the same reason. The expansion of the trustworthiness from a Root CA to the single entities is called chain of trust.

Elliptic Curve Integrated Encryption Scheme

Whenever two communication partners know and trust the public keys of each other but do not possess a symmetric encryption key for confidential communication, they can exploit ECIES [IEE04].

> *ECIES allows two parties with the help of their trusted public keys to*

agree on a symmetric encryption key.

The initiator of ECIES needs besides his own key pair the public key of the receiver. From this information he derives a key. This key is applied to encrypt a symmetric encryption key and create a MAC for validation. Its own public key (V), the MAC (T), and the encrypted key (C) are then passed to the other party. The receiver needs only the provided information and his own private key in order validate the MAC and decrypt the symmetric encryption key. Afterwards both parties can start a confidential communication encrypted with the symmetric encryption key. In this thesis ECIES is always applied as specified for C2X communication in [IEE13b].

Blind Signatures

BLIND SIGNATURES can be defined as follows:

> *When applying blind signatures, the creator disguises the content of the message to be signed before forwarding it to the signer. Therefore, the signer of the message does not know the actual content of the message he is signing. The resulting signature over the blinded message can be however, applied to verify the blinded message as well as the original message.*

Blind signatures were introduced in [Cha82]. When applying blind signatures, the creator and signer of a message are two different entities.

An example use case for blind signatures is digital money. There, the customer sends a blinded message to the bank. The bank then signs this message on reception and deposits the bank account of the customer. Later on the customer can apply the received signature to show that it is valid for the message he possesses to a shop owner and pay for products. The shop owner can then exchange the message with its signature into money at the bank. The bank only knows that it is valid money but is not able to link it to the customer it was issued for.

Ring Signatures

RING SIGNATURES can be defined as follows:

> *A ring signature hides the creator of a digital signature. A verifier cannot distinguish between n possible signers.*

In order to create a ring signature, the signer takes his private key and the public keys of $n-1$ others. Afterwards, the signer sends the signature together with its own public key and the $n-1$ public keys of the others to the verifier. The verifier takes all public keys as input in order to verify the signature. As result the verifier learns if the signature is valid or not. However, the verifier cannot identify the signer of the message. He only learns that the signer is in possession of at least one private key corresponding to an applied public key, but not which. Therefore, ring signature are

applied to hide the identity of the signer in a group of possible signers. Ring signatures were introduced in [RST01]. In this thesis we exploit the variant proposed in [LLZ+07]. This variant is based on ECC, which is also applied in C2X communication. Therefore, IVSs will feature hardware, which allows fast calculations of elliptic curve operations.

Anonymous Credential Systems

ANONYMOUS CREDENTIAL SYSTEMS were introduced by Chaum in [Cha85]. They can be defined as follows:

> *Anonymous credential systems are the digital equivalent of an id, bank, or library card. These cards have in common that they only reveal the attributes necessary for the purpose.*

An id card consists of information like the name of the holder, place and date of birth, and address. A bank card only reveals the name and banking account of the holder, while a library card might only reveal the library id of the user besides his name.

An anonymous credential system allows an entity to decide which attributes are revealed to a verifier, called partial information disclosure. A trustworthy issuer provides an AC to all entities in the system. These ACs denote a set of attributes certified by the issuer. An attribute consists of a key and a value. Possible attributes for an IVS are the *color*, *type*, *brand*, *production date*, and features and sensors like an *electric engine*, *traffic sign recognition*, or *Bluetooth*.

With this AC the entity can prove to a verifier that he or she possesses certain attributes certified in the credential. To convince the verifier, the entity derives a token that only contains the subset of attributes it wants to prove, called minimum information disclosure. So, the verifier learns only the necessary attributes of the entity. An entity can also prove that the content of a cipher text produced for a third party is the value of a certain attribute without revealing the attribute to the verifier. It is possible to prove predicates like *or*, *and*, *greater than*, *smaller than* and *equals* over the attributes. ACs also provide multi-show unlinkability. Several proofs of the same attributes cannot be linked to the same entity. Each time another token is being derived.

As an example, Equation 2.1 shows how one may check if an IVS is manufactured by GM, but without leaking the brand of the IVS. This is realized by checking for all GM brands, if the attribute *brand* equals to the name of the brand. The derived token then indicates, if the equation is true or false and therefore reveals only the manufacturer and not the brand to the verifier.

$$brand = \text{"}Opel\text{"} \vee brand = \text{"}Chevrolet\text{"} \vee brand = ... \tag{2.1}$$

Equation 2.2 can be applied to prove that an IVS is not older than x days. From the result a verifier learns that the IVS is manufactured less than x days ago, but not the exact date.

$$production\_date > today - x \tag{2.2}$$

So called accumulators can be exploited to enable revocation for Anonymous Credentials [CKS09]. An accumulator stores the identifiers of all valid ACs. If an Anonymous Credential shall be revoked, the identifier of this AC is removed from the accumulator. Each time an entity creates a proof it also proves that the identifier of its AC is stored in the accumulator.

Two different implementations of anonymous credential systems exist. U-Prove from Microsoft [PZ13] and idemix from IBM [CVH02]. In this thesis we exploit idemix. It supported at the time of implementation multi-show unlinkability and selective information disclosure, which is necessary for the considered use case. However, U-Proove did not support this features.

2.3. Car-to-X Communication

In future vehicles will not only get information from central servers.

> *In addition to the information obtained from central servers, vehicles will exchange information directly with nearby vehicles, called Car-to-Car (C2C) communication, and the infrastructure like traffic lights or traffic signs, called Car-to-Infrastructure (C2I) communication. If vehicles communicate with both, other vehicles and the infrastructure, it is called C2X communication. Systems that allow this kind of communication are called ITS. Each vehicle participating in the system is called IVS, the server of an application ICS, and communicating infrastructure ITS Roadside Station (IRS). If all of the stations - vehicle, roadside, or central - are meant it is simply called ITS station.*

The data exchanged between IVSs include their current position, direction, and speed [ETS11a]. Also warnings about, e.g., hard breaking vehicles, slow vehicles, or traffic jams ahead are exchanged [ETS10b]. The information exchanged with IRSs includes the phase and timing of traffic lights [CEN15b] and information about speed limits [CEN15a]. The thereby created ad-hoc network is called Vehicular Ad-hoc NETwork (VANET). Because of the high mobility of the IVSs the topology of the network changes frequently. Whenever IRSs are in communication range, they are part of the ad-hoc network.

The infrastructure does not only consist of single IRSs. In fact, multiple IRSs can be grouped into a so called IRS Network. All IRSs within this network are connected

with each other to exchange data. Furthermore, the IRSs can be connected to the Internet.

The applications of C2X communication can be safety or non-safety applications. Examples for safety applications are the traffic jam ahead warning, wrong way driver warning, weather hazard warning, or emergency electronic break light [ETS09b]. Examples for non-safety applications are the reservation of parking lots, payment of toll, or music streaming. For applications which do not require a short time delay mobile communication like Universal Mobile Telecommunications System (UMTS) or LTE might also be exploited.

The communication between IVSs and between the IRSs and IVSs is based on IEEE 802.11p [IEE10]. There exist dedicated communication channels for safety and non-safety applications as well as for control information.

The communication is called DSRC in North America [AST10]. In Europe this name was already introduced for tolling [CEN04]. Therefore, the communication is named according to its frequency band ETSI ITS G5 A/B [ETS09a] whereas A stands for the channels applied for safety applications and B for non-safety applications respectively.

The applied network to application layer standards also differ between the regions. In North America the layers above IEEE 802.11p are specified by the IEEE 1609 family as Wireless Access in Vehicular Environments (WAVE) [IEE13a]. The applications are standardized in SAE J2735 [SAE16a]. For non-safety communication IPv6 [DH98] with TCP or UDP [Pos81][Pos80] can also be applied. The development of the standards is driven by the Crash Avoidance Metrics Partnership (CAMP).

In Europe different standards for C2X communications are substantially created by European Telecommunications Standards Institute (ETSI). They define the so, called GeoNetworking [ETS10c]. Within this, different transport protocols like BTP [ETS14b] and extensions for IPv6 [ETS11c] might be exploited. The applications are standardized individually [ETS11a][ETS10b][ETS10a]. For identification, each application has a unique Application ID (AID). The further development is driven by the Car-2-Car Communication Consortium (C2C-CC).

The different communication stacks for north America and Europe are illustrated in Figure 2.2. This thesis focuses on the European standards. However, this does not mean that the proposed schemes cannot be applied in North America. Furthermore, if we talk about DSRC, we mean the C2X Communication and not tolling.

In the remainder of this section we first describe the structure of GeoNetworking. Afterwards we describe the format of the applied digital certificates. Then, we describe the structure and working of an PKI for C2X communication. Finally, we briefly describe the important messages standardized for safety communication.

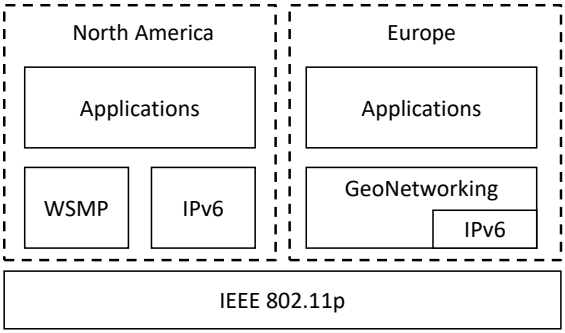


Figure 2.2.: C2X communication stacks in North America and Europe

2.3.1. GeoNetworking

Safety relevant use cases in VANETs like a traffic jam ahead warning or weather hazard warning do not address a single IVS as a recipient of the message. Instead, a geographic region where the message shall be disseminated is specified. If a region is specified as destination, all IVSs in this geographical region are the recipients of the message. For non-safety use cases, it is necessary to address a single receiving IVS. Routing mechanisms in VANETs furthermore have to consider the frequent topology changes caused by the high mobility of the IVSs.

To address these challenges, a routing mechanism called GEONETWORKING has been developed and standardized in [ETS14a].

> *To be identifiable by other entities in the VANET, each IVS has a unique GeoNetworking address. A IVS sending a GeoNetworking message always includes its own GeoNetworking address and geographic position in the message. If the message shall be sent to a single IVS, the last known position and GeoNetworking address of the receiving IVS is also included in the message. When the message is relevant for all IVSs in a region, the geographical region where the message shall be distributed is also encoded in the GeoNetworking message. The sender may also specify a maximum hop limit to reach the receiver.*

The structure of a GeoNetworking message is defined in [ETS14a]. It consists of a *Basic Header* (BH) and a *Secured Packet* (SP). The *Secured Packet* can be further divided in *Header Fields* (HF), *Payload Fields* (PF), and *Trailer Fields* (TF). The *Payload Fields* are composed of a *Common Header* (CH), *Extended Header* (EH), *Transport Protocol* (TP), and the actual *Payload* (P) of the GeoNetworking message. This message structure is illustrated in Figure 2.3.

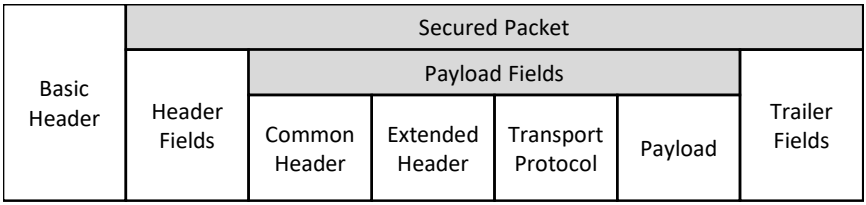


Figure 2.3.: GeoNetworking message format

A more detailed view of the message format is given in Figure 2.4. It shows the detailed message structure for all parts, except the *Extended Header*. The *Extended Header* defines, if the message is Unicast or Broadcast. We describe the meaning of each filed in the following. Afterwards we present the detailed structure of the *Extended Header* for Broadcast and Unicast messages.

The *Basic Header* specifies the version (*VERSION*) of the GeoNetworking protocol, the next header (*NH*), the lifetime of the packet (*LT*) and the remaining hop limit (*RHL*). For secured GeoNetworking messages the next header is always set to 2.

The *Secured Package* [ETS13] starts with the *Secure Package Information* (SPI), which includes the protocol version (*VERSON*) followed by the applied security profile (*PROFILE*) and length of the header fields (HF LENGTH). The security profile defines the contents of the header, payload and trailer fields. Three profiles defined by the standard. One for Cooperative Awareness Messages (CAMs), one for Decentralized Environmental Notification Messages (DENMs), and one generic profile. All of them contain payload and a signature as trailer field. Figure 2.4 shows the generic profile. In the *Header Fields* it consists of the certificate of the sender (*SIGNER INFO*), the generation time of the message (*GENERATION_TIME*), and the generation location (*GENERATION_LOCATION*) of the sending IVS. To correctly parse the *Payload Fields*, the length of the fields and the type of each single field is given as *Payload Field Information* (PFI) first. The payload type can be either *unsecured*, *signed*, *encrypted*, *signed_external*, or *signed_and_encrypted*.

The *Common Header* is applied as standardized in [ETS14a]. The next header (*NH*) is in general set to the *Basic Transport Protocol* (BTP-A) [ETS14b]. The header type (*HT*) defines the routing of the message which corresponds to the type of the *Extended Header*. The header sub type (*HST*) determines the shape of the dissemination area defined in the *Extended Header*. It can be defined according the application requirements as a circle, rectangle or ellipsoid. The *Common Header* also includes the applied traffic class (*TC*), different flags (*FLAGS*), the length of the GeoNetworking payload (*PL*) and the maximum hop limit (*MHL*).

The *Extended Header* can have different type. As mentioned previously, each of

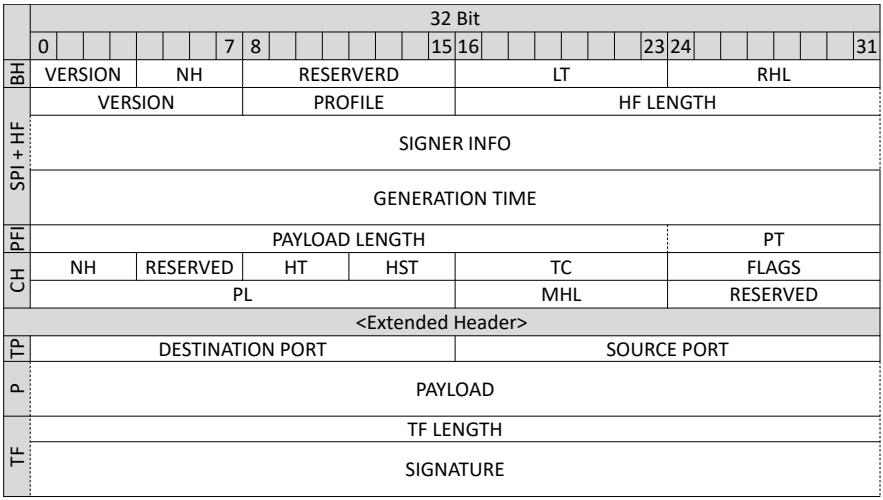


Figure 2.4.: Detailed GeoNetworking message format, without Extended Header

them as its own structure. We describe the structure of the Geographically-Scoped Unicast (GUC) and Geographically-Scoped Broadcast (GBC) later.

For the *Transport Protocol* different choices such as the Basic Transport Protocol (BTP) and IPv6 exist. However, in general the BTP is applied as illustrated. It consists of a source (*SOURCE_PORT*) and a destination port (*DESTINATION_PORT*). The Payload (*PAYLOAD*) contains the actual content of the message. The concluding *Trailer Fields* contain the length of the trailer fields (TF LENGTH) and a signature (*SIGNATURE*) over the complete *Secured Package*. The signature is applied using ECDSA with National Institute of Standards and Technology (NIST) curve P256 and SHA-256 as hash function.

Geographically-Scoped Unicast

The content of the *Extended Header* of a GUC message is given in Figure 2.5. It contains of a sequence number (*SN*), to detect duplicate GeoNetworking packets and indicate the index of the sent packet. Furthermore, it contains the position of the source (*SO PV*) as *Long Position Vector* and the destination (*DE PV*) as *Short Position Vector* [ETS14a].

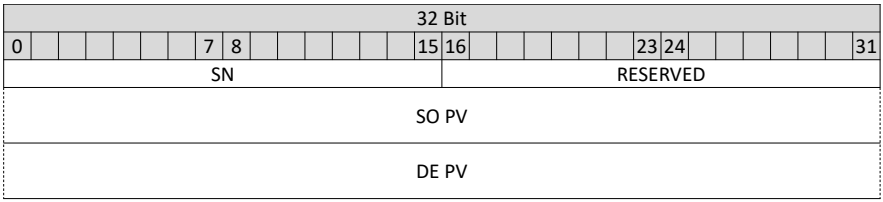


Figure 2.5.: Detailed structure of the Extended Header of a GUC message

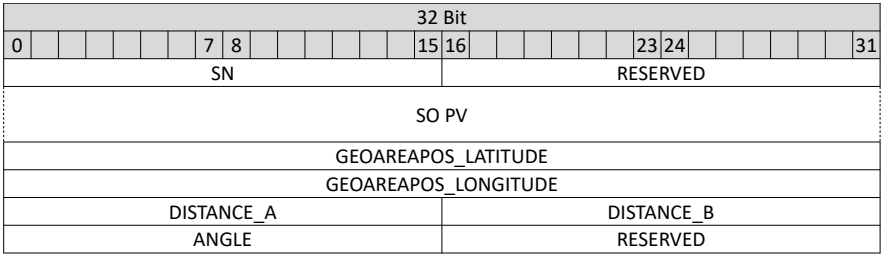


Figure 2.6.: Detailed structure of the Extended Header of a GBC message

Geographically-Scoped Broadcast

The *Extended Header* of an GBC message also contains a sequence number (*SN*) and the position of the source (*SO PV*) as Long Position Vector. It furthermore consists of the description of the geographic area the message shall be distributed as defined in [ETS11b]. The area is defined by the latitude (*GEOAREAPOS_LATITUDE*) and longitude (*GEOAREAPOS_LONGITUDE*) of its geographic center, two distances (*DISTANCE_A* and *DISTANCE_B*) to describe the size, and a rotation angle (*ANGLE*). This structure is illustrated in Figure 2.6.

2.3.2. Certificate Format

ITS stations exploit a special certificate format [ETS13] for C2X communication over DSRC. Because of the limited channel capacity, it is optimized to have a small size.

Three different profiles for certificate types are defined in [ETS13]. The types are ATs, Enrolment Credentials (ECs), and CA certificates. ATs and ECs are exploited by IVSs. The CA certificates are used by the different CAs of the PKI.

ECs are valid for a long time period. They are like an identifier of an IVS. Normally this certificate is generated at production. It is never exploited in communi-

cation with other ITS stations. It is only used to request ATs from the PKI. In the request, it is applied as the proof to be eligible to obtain ATs. The ATs are exploited to communicate with other ITS stations. To prevent tracking, they are only valid for a short time period and changed frequently.

CA certificates are used by the different entities of the PKI to ensure their role. In the next section the single entities of the PKI and how an IVS obtains its EC and ATs are described.

A certificate consists of a *version information*, *signer information*, *subject information*, *subject attributes*, *validity restrictions*, and a *signature*.

The *version* specifies the version of the certificate. The *signer information* defines the issuer of the certificate. They can be self-signed or contain the hash of the issuing certificate.

The *subject information* specifies the type of the certificate. If it is an EC or AT of an IVS, an CA certificate from an CA or a signer of a Certificate Revocation List (CRL).

The *subject attributes* contain the properties of the ITS station. This includes the keys for signing and encrypting messages, the assurance level, a list of AIDs and a list of Service Specific Permissions (SSPs) for the referenced AIDs. The assurance level defines how good the platform and private keys of the ITS station are protected. Furthermore, it also specifies the confidence of this information. If the certificate is of type AT, the list of AIDs indicates the applications which can be exploited with this certificate. If it is from the type EC, it defines the applications an IVS can obtain ATs for. Whenever the certificate is from the CA type, it defines the applications these authorities can issue certificates for. The list of SSPs defines the permissions for each of the applications.

The *validity restrictions* limit the geographic area and period in time a certificate is valid. The last part of a certificate is a digital signature over all information created by the issuer. As signature algorithm ECDSA with the NIST curve P256 is applied.

2.3.3. Public Key Infrastructure

An PKI for C2X communication in Europe consists, as illustrated by Figure 2.7 of a Root CA, an Authorization Authority (AA), and an Enrolment Authority (EA). The Root CA is the trust anchor. The EA issues the EC to the IVSs. This is done at time of production. The AA issues the ATs to the IVSs. This process is repeated regularly. ATs are only valid for a short period of time.

Different ways actually exist aimed to obtain the outlined ATs in order to send safety relevant messages over ETSI ITS-G5A. One of the ways was developed by the C2C-CC [BSS+11]. Another has been standardized by ETSI [ETS12b]. Also more complex variants offering special features [KJP14] [WWKH13] have been pro-

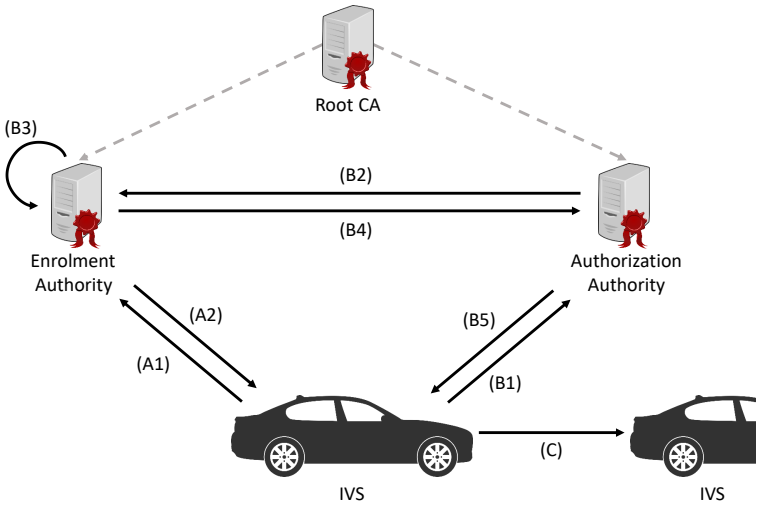


Figure 2.7.: PKI applied in C2X communication

posed. However, in all variants the IVS first requests an EC from the EA and later on applies it to obtain ATs from the AA. We only discuss the former one, as the concept we propose in Chapter 4 can be seen as an extension. Figure 2.7 illustrates the single steps an IVS must complete prior to sending safety relevant messages. In the following these steps are discussed in more detail.

Enrolment Certificate Request

First, the IVS has to request an EC from the EA. This is done in Step A1, where the IVS sends an encrypted and signed message with its canonical certificate together with a certificate request to the EA.

$$Enc_{EA}(Sig_{IVS}(canonical\,cert., cert.\,request)) \tag{A1}$$

Then, the EA validates the request and responses with the encrypted EC (Step A2). After reception, the IVS decrypts, validates, and stores the EC.

$$Enc_{EC}(EC) \tag{A2}$$

Authorization Tickets Request

After an IVS received its EC, it requests as illustrated in Figure 2.7 ATs at the AA. To request ATs, the IVS first generates key-pairs to be certified, signs the corresponding

Public Keys (PKs) with its EC, encrypts the resulting signature and the EC for the EA, and sends the PKs together with the encrypted payload to the AA (Step B1). The EC is encrypted in order to hide the identity of the IVS from the AA. Furthermore, the signature is created and encrypted in order to allow the EA to verify that the PKs belong to the request and the encrypted EC is not just replayed.

$$PKs, Enc_{EA}(Sig_{EC}(PKs), EC) \tag{B1}$$

Upon reception, the AA calculates the hash of the PKs and sends the hash value together with the encrypted EC and signature over a secured channel to the EA (Step B2).

$$H(PKs), Enc_{IVS\_EA}(Sig_{EC}(PKs), EC) \tag{B2}$$

Next, the EA decrypts the signature and EC to validate them (Step B3). Afterwards, it sends the result of the validation back to the AA (Step B4 depicted in Figure 2.7).

$$OK\ or\ FAIL \tag{B4}$$

If the result is positive, the AA issues the ATs to the IVS (Step B5). Otherwise, an error is returned.

$$ATs\ or\ Error \tag{B5}$$

After successful execution of these steps, the AA has knowledge about the ATs issued to the IVS, but not its identity. On the other side the EA knows the identity of the IVS, but not the issued ATs. Therefore, the knowledge of both entities is necessary in order to link an AT to a certain IVS.

Authorization Tickets Usage

After the IVS received the ATs, it can start broadcasting signed messages for safety applications over ETSI ITS G5A. This is illustrated by Step C in Figure 2.7. When the validity period of the obtained ATs expires, the IVS needs to request new tickets from the AA to continue sending valid messages.

An IVS is in possession of multiple ATs which are valid at the same time. To protect its privacy an IVS frequently changes the AT it applies. Therefore, an attacker who records the exchanged messages of IVSs is not able to link messages of the same IVS whenever the AT is changed in between. Because of this feature ATs are also called pseudonyms. In this thesis bother terms are applied as synonyms.

2.3.4. Messages

The most common messages in European VANETs are the CAM, DENM, and Service Announcement Message (SAM).

The CAM [ETS11a] is send periodically, up to 10 times a second. The frequency depends on the speed and heading of the IVS. It contains general status information about the sending IVS like position, speed, and heading. CAMs are not forwarded if received. They are always single hop messages.

DENMs [ETS10b] are event based. They warn the IVS about upcoming dangerous situations, like the end of a traffic jam, hazardous weather events, hard breaking IVSs, or wrong way drivers. The content of the message are the event type and areas where the message shall be distributed and is relevant. These messages can be multi hop messages, which are forwarded by receiving IVSs.

SAM [ETS10a] messages are sent periodically in order to inform nearby ITS stations about its services. It contains the list of services, offered by the sending ITS station.

2.4. Multimedia Broadcast Multicast Service

The current high speed communication standard for mobile networks is called LTE [3GP13a]. End user devices like smartphones and tablets are called User Equipment (UE) and connect to the base stations of the mobile network, called evolved Node B (eNodeB). The eNodeB itself is connected to the core network of the Mobile Network Operator (MNO).

Over this network it is able to reach the Internet. This communication path is normally exploited when the UE exchanges data with entities in the Internet. All these connections are normally unicast, where one UE exchanges data with one server located in the Internet.

For some use cases unicast is not suitable. Several use cases require the transmission of data to all or a group of UE. Whenever data shall be sent via broadcast or multicast from a server in the Internet to all or a group of UEs MBMS can be applied [3GP13b]. In the following we first describe the general working of MBMS, followed by its architecture.

Each Content Provider aiming in distributing data to a group of or all UEs has to register a MBMS User Service for each application it offers first. Afterwards, UEs register for each User Service they are interested in. The data of a User Service is distributed in one or multiple predefined MBMS service areas. Each of these areas can consist of multiple eNodeBs. MBMS User Services are not available continuously. Each session is advertised by a service announcement. Before the data is transmitted a session start indicates that data is ready. A session stop is sent to the UE when there is no more data to transmit. If a User Service is sent to UEs in multi-

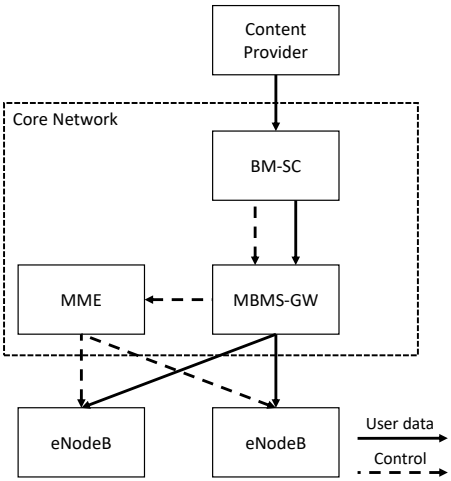


Figure 2.8.: Architecture of MBMS

ple service areas, different content can be distributed each of these areas. Then, one session needs to be initialized for each different content. However, these areas cannot overlap. The data provided by the content provider is sent once to all registered UEs. MBMS supports traffic classes for background data, like software updates, and streaming.

Figure 2.8 illustrates the entities involved in the MBMS data distribution and their connections. These are namely the Broadcast Multicast Service Center (BM-SC), Mobility Management Entity (MME), and MBMS-Gateway (MBMS-GW) within the core network of the MNO. Furthermore, a Content Provider is necessary to provide the data to the core network and eNodeBs which distribute the data to the UEs.

The Content Provider sends its data to the BM-SC. On reception the BM-SC processes the data. The BM-SC then sends control information like session start and stop and the area along with the data to the MBMS-GW. The MBMS-GW splits up the data and control information. The control information is forwarded to the MME, which sends it to the responsible eNodeBs. The data is directly forwarded from the MBMS-GW to the eNodeBs. We assume each eNodeB has an integrated Multicell Coordination Entity (MCE).

3 | Related Work

We review the related work in this chapter before proposing our schemes in the following chapters. We describe previous works in the field of anonymous credentials, key agreement protocols, path hiding, and geocast.

3.1. Anonymous Credentials

ACs are exploited for various use cases. In the sequel we outline vehicular as well as general applications.

PUCA [FKL16] exploits an AC system to request ATs for safety related communication from the AA. In the scheme an IVS first obtains an AC from the EA and uses this AC later on to request unlinkable ATs from the AA. To revoke an IVS, dynamic accumulators and periodic no-show credentials are applied. Periodic no-show credentials can be exploited at almost n times per time period. This prevents an IVS from requesting more ATs in a time period than allowed by the system. However, PUCA only changes the way of obtaining new ATs for safety-related communication and how to revoke them. It does not consider ATs for non-safety applications. An illustration of the scheme is given in Figure 3.1.

Singh [Sin12] applies anonymous credentials to anonymously authenticate messages in C2C communication. He utilized idemix for the implementation and proposed two versions. The first one does not support revocation. However, the credentials are only valid for a short period of time. The second version does not require short term credentials for revocation. Instead, they exploit accumulators for revocation. To ensure the same credential structure for all IVSs, they are obtained from a central government organization. However, the author applied the scheme only for safety messages, which are directly exchanged between IVSs. Furthermore, the overhead to authenticate a message introduces a too large time delay for safety messages. Moreover, unlinkability between individual messages is undesirable for safety use cases like intersection collision warnings.

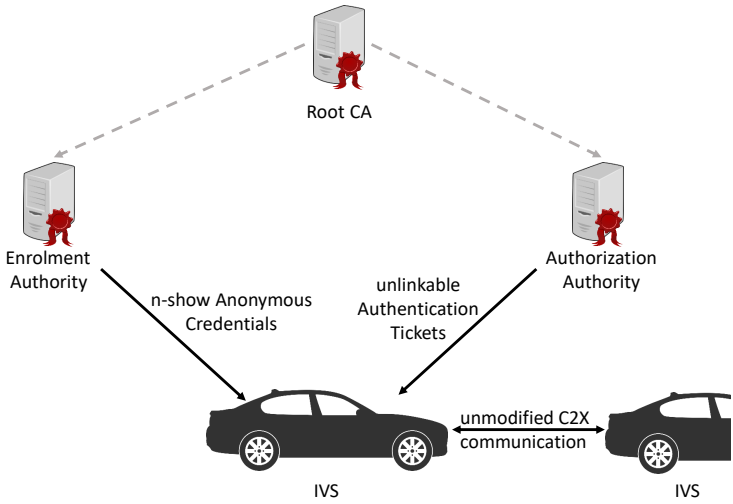


Figure 3.1.: PUCA scheme

A protocol to enhance the privacy of electric vehicles communicating with the charging infrastructure was proposed in [HPSK13] based on ACs and other cryptographic schemes to extend the standardized protocol. The identity is kept private by exploiting anonymous credentials to store charging contract details like the expiration date.

The privacy-friendly smart environment proposed by Armac et al. [APPR09] also exploits idemix to build an anonymous credentials system. The system uses a mobile based application to extract user preferences for, e.g., lighting and favorite music, when switching between different smart environments like home or hotels. Users obtain an anonymous credential in order to authenticate for and use the different environments. For each visited environment, only the necessary information is disclosed. Therefore, this approach prevents the tracking of users in smart environments. Requests of the same user at different visited locations cannot be linked.

Aimeur et al. introduced a privacy preserving e-learning system in [AHO08]. It consists of a set of protocols exploiting anonymous credentials in order to preserve the privacy in e-learning environments. They propose protocols for, e.g., course registration or proofs for transcripts and degrees.

Different government funded projects like PRIME [PRI], ABC4Trust [ABC], and FutureID [Fut] also show possible fields of application. They applied ACs to, for example, allow an anonymous participation in an online courses evaluation at an

university or limit the access to information shared within social networks.

3.2. Anonymous Key Agreement and Authentication Protocols

In Chapter 5 we propose an anonymous authenticated key agreement protocol which allows two IVSs to agree on a symmetric encryption key and prove their membership of a certain group without revealing their identity. Furthermore, it reduces the number of necessary ATs. In this section we review the related work.

The Institute of Electrical and Electronics Engineers (IEEE) standard for security in VANETs [IEE13b] applies ECIES in order to authenticate and encrypt the communication between two IVSs. First the IVSs agree on a symmetric encryption key by deriving it from their asymmetric keys embedded in the pseudonyms. This key is then used to encrypt the messages exchanged between the IVSs. If an application exploits ECIES, it has its own set of application-specific pseudonyms to agree on a symmetric encryption key. To prevent tracking, the IVSs have to change all their application-specific pseudonym at the same time. Therefore, each application needs the same amount of pseudonyms. If an IVS possesses less pseudonyms for one application, some of them will be applied with different pseudonyms of other applications. An attacker can then easily link the two different pseudonyms. An example for this is given in Figure 3.2. In the upper part the vehicle only changes one of its application-specific pseudonyms with the pseudonym for safety-related communication. Therefore, an attacker can easily link *Pseudonym 1* to *Pseudonym 2* because they were both used with the same *Pseudonym A1* from *Application A*. Furthermore, the attacker can also link *Pseudonym 3* to the same vehicle, because *Pseudonym B2* was used also in combination with *Pseudonym 2*. When all pseudonyms are changed at once, as illustrated in the lower part of Figure 3.2, it is not possible to link the different pseudonyms to the same vehicle. The key agreement protocol proposed in Chapter 5 reduces the amount of necessary application-specific pseudonyms in contrast to a simplistic use of ECIES to prevent linking of pseudonyms.

The authors of [LLZ+07] propose ring signatures for anonymous routing in wireless ad-hoc networks. Their protocol applies ring signatures to hide the identity of the entities while agreeing on an encryption key. However, they did not evaluate related ring building strategies nor the size of the protocol nor multiple own pseudonyms. In [FRH09] the authors advocate ring signatures in mobile ad-hoc networks for privacy preserving authentication of neighbor nodes. They did investigate ring building strategies, but most of their strategies require either a central server or the nodes have to be a-priori aware of the pseudonyms of all other nodes. In addition, these authors only considered the case where each node has just one pseudonym and elaborated

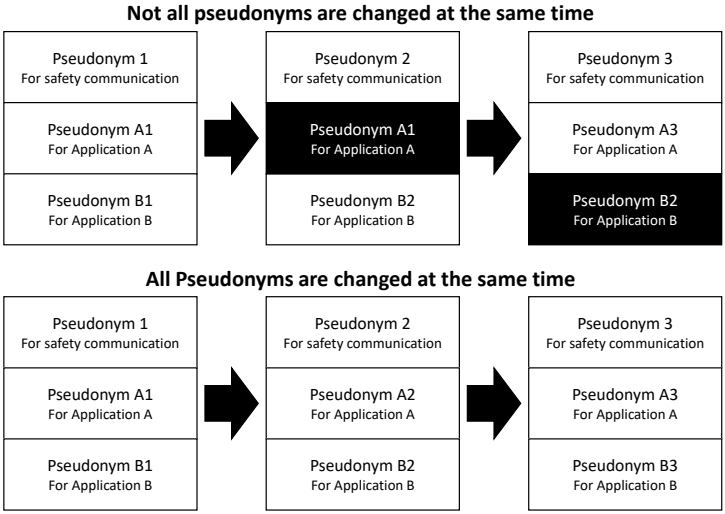


Figure 3.2.: Linkage by unchanged pseudonyms

a general formula to calculate the transmission overhead, but they did not evaluate their suggestion.

If two parties aim to secretly identify whether they belong to the same group, then they can use Secret Handshakes [BDS+03]. Secret Handshakes have the property that even if the handshake fails, none of the communicating parties or an eavesdropping third party can discover the group of the communicating parties. A third party is thus never able to identify the group of the two communication entities, even if the protocol is executed successfully. To prevent an attacker from linking two different handshakes, both communicating parties have to use a different pseudonym for each handshake. Therefore, this scheme is similar to ECIES, but it has the additional property that a third party is never able to unveil the group. However, group affiliation is no secret in C2C communication. IVSs broadcast the groups they belong to periodically in SAMs to all other IVSs in communication range. The aim of the protocol evaluated in Chapter 5 is to reduce the number of ATs necessary for each application. Secret Handshakes are not suitable for this purpose because the same amount of ATs is necessary as when ECIES is exploited.

When applying Group Signatures [CH91], a set of entities form a group. Each member of the group has its own private key, which can be used to sign a message on behalf of the group. In addition, there exists a public group key which can be used to verify signatures created by group members. The verifier of a signature is

not able to determine which member of the group created the signature. Each time a member leaves the group information has to be distributed to all remaining members of the group. This requirement obstructs the scaling in the considered use case where vehicles frequently leave the group. In addition, one cannot assume that every IVS has a constant online connection which is a prerequisite to get this information from a central entity.

The Enhanced Privacy ID (EPID) from Intel [BL10] extends Direct Anonymous Attestation (DAA) [BCC04] to anonymously authenticated devices. The system has three roles: issuer, member, and verifier. The issuer creates the, possible blinded, keys for the members and delivers it to them. The member applies its key to convince the verifier that he is a member of the claimed group. In order to verify the membership, the verifier applies a public group key. Therefore, this scheme is similar to group signatures.

With Matchmaking Protocols [BG85] two members of the same group can authenticate each other without leaking the group they belong to. However, this method reveals the identity of the communicating entities. We, in contrast, aim to hide the identity of the partners and not of the groups they are members of. Therefore, this method is not suitable in our context.

In [KWC14] the authors propose an anonymous authentication scheme for mobility networks. They assume a mobile user who is registered to a home agent aiming in exploiting the roaming service of a foreign agent. However, in order to work, their scheme assumes a pre exchanged secret between the user and its home agent in order to establish a secure authenticated channel. IVSs in VANETs normally meet IVSs they never met before and unlikely meet again. Because of the huge number of IVSs it does not scale to agree on a secret which each possible IVS beforehand. Therefore, such schemes are not suitable to anonymously establish a secure authenticated channel in C2C communication.

The authors of [CLLW08] exploit ring signatures and blind signatures to authenticate a client at a server. Their protocol hides the identity of the client to the server, whereas the server reveals its identity to the clients. However, in our use case both communication partners aim at hiding their identity.

Physical Unclonable Functions (PUFs) can be used to generate private keys. According to [FPK13] this reduces the amount of necessary secure storage to save the keys. However, the number of ATs remains the same. In contrast the investigated protocol in Chapter 5 reduces the number of ATs. This also reduces the amount of secure storage, because less keys have to be saved. In addition, less ATs have to be generated by the infrastructure, less have to be sent to the IVSs and less have to be saved. However, PUFs can be used in addition to further reduce the amount of necessary secure storage.

Figure 3.3.: Example of periodic changing pseudonyms

3.3. Path Hiding

Different methods exist to hide the traveled path of entities. In this section we outline some of these methods.

Pseudonyms as used for safety communication in VANETs can be applied to hide the identity of a IVS. The IVS then changes its identity on a regular basis. Therefore. the IVS will feature different identities at the start and end point. An example where an IVS is driving from *Mainz* to *Rüsselsheim* is given in Figure 3.3. The identity of the IVS is *black* when it starts in *Mainz* and is *green* when it arrives in *Rüsselsheim*. It is not possible to link the two identities together. An observer in between would even record *blue* or *orange* as the identity of the IVS. However, this is only effective, if the attacker does not have access to all data of the IVS. In a scenario where all data is sent to an IVS, an attacker has access to all data along the traveled route. Therefore, this data can be exploited to reconstruct the path and sensitive locations like the home or the workplace of a driver as demonstrated in [Kru07].

The authors of [CGR+11] propose path hiding strategies based on the exchange of sensor data. They propose different strategies on how to exchange the data. However, they do only consider participatory sensing applications for smart phones. Therefore, they do not take the special characteristics of IVS into account. IVSs in a VANET have a communication range of several hundred meters. The range of Bluetooth used for direct communication between smart phones is much less. IVSs also move along roads and meet each other more frequently.

As detailed in [Kru07] spatial cloaking can be used to drop all sensor samples near sensitive locations. Then, only the data in non-sensitive locations is reported

to the central server. However, this can be critical in, e.g., residential areas. If all drivers define their home as sensitive locations, rarely an IVS uploads it collected data.

In [HG05] the authors consider a traffic monitoring system, where the IVSs periodically report information like speed and location to a server. To confuse the attacker, they modify the positions reported by different IVSs in such a way that it looks like they crossed their path. However, adding an error to the position samples sent to the service provider is not tolerable for use cases like the collection of potholes or traffic signs. Other methods [HKH10] add Gaussian noise to the reported location, which also introduces an unacceptable error.

SLICER was proposed in [QWC13] and [QWC15]. It splits each sensor reading in a number of slides. These slides are then distributed to other participants, which upload the slides to a central server. If the central server receives at least k slides of the same sensor reading, it can reconstruct the data. They propose two different strategies, the first one just slices the data, distributes it to the other participants, which upload the data. In the second version the originator of the data predicts the participants it will meet until a given time frame and slices the data accordingly to meet time requirements and minimizing the total costs.

A system to ensure a minimum k-anonymity for vehicles reporting data to an ICS is presented in [FLK15]. The authors apply a distributed secret sharing algorithm with location and time specific keys to accomplish this. They exploit DSRC in order to establish the location and time specific keys. These keys are then reported by means of a secret sharing algorithm. Whenever k vehicles reported data encrypted with the same key, the ICS is able to decrypt the data. However, the accuracy of the location and time of the data to report needs to be reduced in order to reach the desired indistinguishability. This is not acceptable for high-precision maps. Furthermore, if the accuracy is high enough to uniquely identify a specific sensor event, the vehicles can aggregate it immediately by exploiting DSRC.

3.4. Geocast

In this section we review the related work on geocast for ITS applications.

In [JRX11] the authors propose a Grid Based Geocasting Scheme (GBGS) for ITS applications. They divide the surface of the world into tiles in order to define possible dissemination areas. The size of each tile is adjusted according to the number of IVSs within. When more IVSs than a threshold value are present in a tile, it is subdivided into two tiles of equal size. If the number of IVSs in two neighboring tiles drops below another threshold value, then they are merged. Each IVS is aware of the tile it is currently in. Every time an IVS leaves a tile, its current position is transmitted

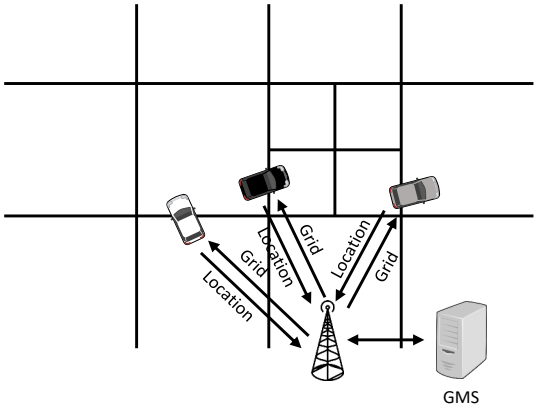


Figure 3.4.: Illustration of the GBGS

to a so-called Geo Messaging Server (GMS). On reception, the GMS determines the new tile the IVS is located in and sends it back to the IVS. Therefore, the GMS is all the time aware of the position of all IVSs. This procedure is illustrated in Figure 3.4. An ICS aiming to send a message to each IVS in a geographic area needs to query the GMS for all IVSs in the dissemination area first. The server then determines and returns all IVSs located in the corresponding tiles. The disadvantage of this scheme is clearly the central GMS, which is aware of the coarse position of all IVSs and is therefore able to track them and thus may infringe their privacy. Furthermore, the scheme does not scale because each message has to be distributed to each IVS via a single unicast message. In addition, this scheme does not support the addressing of a group of IVSs in the first place. However, this feature was later on added as part of the COmmunication Network VEhicle Road Global Extension (CONVERGE) project [CON15a]. In the evaluation section for Chapter 6 we compare this scheme to our Anonymous Geocast scheme for ITS Applications (AGfIA) approach.

In LTE MBMS [3GP13b] can be exploited to distribute data from a content provider to a group of recipients in predefined service areas by means of multicast. In order to apply MBMS, each application has to register an MBMS User Service at the MNO first. An IVS aiming to exploit several applications has to register for each application separately. MBMS was developed to download a huge amount of data or to stream audio or video data from a radio or TV station to many recipients. For this reason, it is based on multicast in order to save bandwidth. Therefore, this scheme is not well-suited to distribute the rather small ITS messages. In order to support the distribution of different messages in various service areas, one MBMS session has

to be initiated for each service area, but this introduces a high complexity. Furthermore, it is not possible to have overlapping service areas. In addition, messages are not repeated automatically in order to inform IVSs entering the service area. Consequently, the messages have to be sent periodically from the content provider to the MNO, which spreads them in the service area. Obviously, this method is not very efficient. We compare this scheme with AGfIA in Chapter 6.

The authors of [CMGK14] analyze the LTE unicast and MBMS transmission modes for safety-related ITS applications. They further study the configuration of MBMS for safety-related ITS applications. Their proposed configuration consists of a central entity which receives all messages. It is accessible by all MNOs and distributes the messages via all mobile networks covering the dissemination area. The authors also state that a new data delivery method for MBMS is necessary to fulfill the requirements of ITS messages. They conclude that MBMS is more efficient in terms of resource consumption when compared to unicast messages. However, this seems obvious, because less messages have to be transmitted in multicast compared to unicast. Furthermore, they do not consider multiple ITS applications with different subscriber groups.

The transmission of ITS messages via LTE and MBMS has also been studied in [ACC+13], [ETS12a], and [VRBZ08] but none of these works provides a solution which fulfills all requirements of ITS applications. Unicast communication via LTE does, for example, not scale for a large number of vehicles and for MBMS a signaling overhead to manage all the different receiver groups is introduced.

In [dMdlIZ15] the authors propose a geocasting mechanism where the data to distribute is sent to only a subset of all vehicles present in the desired region. These vehicles then further distribute this data to nearby vehicles. However, this mechanism requires a central geoserver which is aware of vehicles present in the desired region in order to select a subset. Therefore, it is possible to track the movement of these vehicles.

Three methods of cellular geocast which form the current state of the art were studied in [JRX11]. In the first method a central server, aiming at the distribution of a geocast message, sends an inquiry to all clients requesting their location. From the response, the server selects the relevant clients and sends the message to them. This method clearly does not scale for a large number of clients and features a considerable delay in message delivery. The second method requires all clients to send periodical position updates to a central server which stores them in a database. When a message shall be sent to all clients in a geographic region, the central entity queries its database and sends the message to the relevant clients. An example scheme exploiting this method for C2X communication is given in [FWZ12]. This method does not suffer from the additional delay of the first method. However, it introduces some blur on the position data, because some clients might have moved away since the last

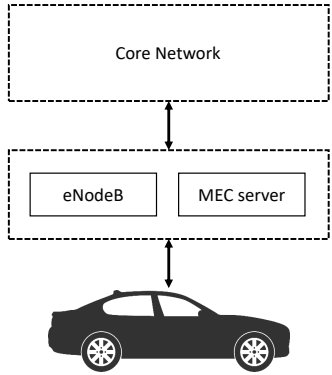


Figure 3.5.: Location of the MEC server within the mobile network

position update. In the third method the clients autonomously update their current location at the central entity when they moved a certain distance. This improves the accuracy of the positions in comparison to the second method. Nonetheless, it still has scalability problems as the first two methods. Last but not least, all these methods do not protect the location privacy of the IVSs. They require regular position updates from the IVS to the ICS.

ETSI published in March 2016 several specifications for Mobile Edge Computing (MEC) [ETS16c] [ETS16b] [ETS16a] containing technical requirements and explanatory material. When applying MEC an application server is located directly at the edge of the mobile network e.g. directly connected to an eNodeB. This placement is also illustrated in Figure 3.5. The concept of MEC works as follows. Whenever an IVS detects an event like an accident it generates a message. This message is then stored directly on the application server at eNodeB level and distributed within the cell. Vehicles entering the cell can then later on receive this information even if no other vehicles are present anymore. Furthermore, it is possible to upload aggregated information to backbone servers. However, currently only a concept is available.

4 | Attribute Based Authentication

The connectivity of current vehicles allows the driver to use convenience services and even install applications. These applications might require authentication and feature access restrictions based on vehicle attributes. A simple mechanism would be that the marketplace or application knows all attributes of the participating IVSs. This however harms the privacy of the IVS which aims at hiding the attributes not necessary for the exploited applications. Therefore, we propose in this chapter a scheme, which allows an ICS to require certain attributes of subscribed IVSs in order to successfully authenticate. During the authentication the privacy of the IVS is preserved. Furthermore, the scheme is complaint with the standards for C2X communication. This chapter is a polished and extended version of the publications [BH14] and [BH16b].

4.1. Motivation

In safety relevant communication over ETSI ITS-G5A an IVS authenticates itself by signing all messages with an AT. Besides these safety applications, an IVS can also exploit non-safety applications over ETSI ITS-G5B or cellular communication. Examples of non-safety applications are parking lot reservation or music streaming. To authenticate this kind of communication, it is not sufficient to sign all messages with an AT for safety applications. Some applications may have access restrictions based on attributes. A possible restriction could be, for example that the IVS is of a specific brand or that certain sensors or features have to be present. Some ICSs might also aim at billing the IVS for usage of their application.

Whenever an IVS uses an application with access restrictions based on attributes, like for sensor data uploads, it aims at hiding its identity and the not necessary attributes to protect its privacy of the identity. This is comparable to a person in the real world showing its bank or library card. These cards reveal just the necessary personal information. If the not required attributes are also revealed, an application

might exploit this information in order to discriminate the driver of the IVS by charging higher prices for the same service when the requesting IVS is a more expensive model [HSL+14].

We propose a system that relies on ACs as suggested in [CL01] to issue attribute-based ATs. We apply the ATs as standardized in [ETS13]. To get these ATs the IVS needs to prove with its AC that it possesses the attributes necessary to use the specific application. The proof contains only the attributes necessary to exploit this application. All other attributes are kept private and thus away from the verifier. Each AT can be exploited for the requested application only. Therefore, this protects the privacy of association of the IVS. Furthermore, we describe how the communication can be secured in order to also protect the privacy of communication of the IVS.

Two different versions of the system are proposed in this chapter. OEMs will most likely issue the AC for their manufactured IVSs. When verifying the proof of attributes, it is possible to identify the issuer of the AC and therefore the OEM of the IVS. When applying the first version it is possible to determine the OEM of the IVS whenever it requests ATs because the OEM is the issuer. An IVS might, however intend to hide this information for privacy reasons. In contrast, the second proposed version prevents a verifier from getting knowledge about the OEM of the IVS. Now all ACs used to prove attributes are issued by a Trusted Third Party (TTP). To enable this scheme, the IVS requests from the TTP a second AC with the same attributes as the one issued by the OEM. Therefore, the verifier of the attributes does no longer learn the OEM of the IVS. However, this version does not only introduce a TTP, it furthermore increases the complexity of the system. We compare the pros and cons of both versions in the evaluation section of this chapter.

Moreover, we show how the IVS can pay with digital money if the application requires billing. In addition, our approach supports the revocation of an IVS, ATs, ACs, and single attributes, respectively. The privacy of the IVS to the other entities always stays protected. We created a prototype implementation and evaluated it resulting in the fact that the delay introduced by the anonymous credential system does not affect the usability of applications.

4.2. Entities

We apply the following entities in our system to obtain attribute-based ATs:

Root CA: The Root CA is the trust anchor of the system. It certifies the Enrolment and Authorization Authorities, TTP, and Service Directory (SD). Every IVS in the system has to trust the Root CA first in order to trust the other entities.

IVS Enrolment Authority: The IVS Enrolment Authority (IVS EA) issues the EC and AC to the IVS. In comparison to the previous work we precede it with *IVS*. This is to distinguish it from the EA responsible for ICSs. In order to issue ACs, the IVS EA maintains a database with all attributes of all IVSs it is responsible for.

IVS Authorization Authority: The IVS Authorization Authority (IVS AA) issues ATs to an IVS after it has checked its eligibility. The AT allows an IVS to exploit the application specified in the AT. We precede it, in comparison to the previous work, with *IVS* to clarify that it is responsible for IVSs.

ICS Enrolment Authority: The ICS Enrolment Authority (ICS EA) issues ECs to ICSs. The preceded *ICS* shall indicate its responsibility for the ICSs.

Trusted Third Party: The TTP is exploited for revocation in the second proposed version of the scheme. It stores all eligible IVSs.

Bank: The bank is a central entity which issues digital money. This money may be used by the IVSs to purchase ATs at the IVS AA in order to get access to an application.

Service Directory: To enable an IVS to search for possible applications and facilitate an ICS to advertise its applications, the SD manages a list of all available applications with information such as access restrictions.

ITS Central Station: Each ICS possesses an EC issued by the IVS EA to prove it is an authorized ICS in the system. An ICS can offer one or several applications to the IVSs. Applications can have the condition that the IVS using it must have certain attributes like belonging to a specific brand or presence of a certain sensor. Furthermore, each application has an assigned unique identifier. An ICS can also have its own IVS AA to issue ATs. This IVS AA is then used instead of the global one for the applications offered by the ICS.

ITS Vehicle Station: An IVS requests an EC and AC, containing its attributes from the IVS EA. Later on, it applies them to obtain ATs from the IVS AA. The IVS proves its eligibility for using an application by presenting the AT to the ICS.

4.3. System

In the following, we detail the steps which are necessary for an IVS to authenticate for a ICS featuring access restrictions. They are similar to the steps described

in [BSS+11] because we used them as a foundation of this work. As a prerequisite besides the establishment of the PKI, the information about the attributes an IVS possesses must be stored at the IVS EA. Furthermore, the ICS has to provide the necessary attributes, billing information, etc. to the SD to enable the issuance of corresponding attribute-based ATs.

When an IVS aims at exploiting an application offered by an IVS, it needs to get a valid EC, AC, and digital money first. Afterwards it applies them to obtain attribute-based ATs. These ATs may then be used for authentication at the ICS. We provide two different versions dealing with how this can be established. They differ in the way EC, AC, and ATs are obtained and revoked. We apply the following notation for the single steps: <*Phase*><*NumberOfStep*>. If a step is only executed in a specific version, we append *#1* for the first and *#2* for the second version to the name of the step. Furthermore, steps that are only valid for the first version are illustrated with dotted arrows and steps only valid for the second version are illustrated with dash dot lines in the figures. We assume an encrypted communication by, e.g., Transport Layer Security (TLS) between all entities in the backend. When Y is signed by entity X, it is written as $Sig_X(Y)$. Whenever payload Y is encrypted for entity X, it is written as $Enc_X(Y)$.

4.3.1. Enrolment Certificate and Anonymous Credential Request

First, the IVS must obtain an EC and AC from the IVS EA. This procedure is illustrated in Figure 4.1. The EC is a certificate to prove that the IVS is an authorized ITS station. It is the same EC as in previous work [ETS12b]. The AC is a certification of the attributes of the IVS as proposed in [Cha85]. To get an EC and AC, the IVS sends in Step A1 its canonical certificate and a certificate request signed with its canonical keys encrypted to the IVS EA.

$$Enc_{IVS\_EA}(Sig_{IVS}(canonical\,cert., cert.\,request)) \tag{A1}$$

On reception, the IVS EA checks, whether the IVS is eligible to obtain the EC and AC. Afterwards, the Steps A2#2 and A3#2 are performed, if the second version is applied. In Step A2#2 the IVS EA creates a unique identifier and sends it to the TTP.

$$identifier \tag{A2\#2}$$

There it is saved in an accumulator which stores all valid identifiers and therefore IVSs. Subsequently, the TTP responds in Step A3#2 if the operation succeeded or not.

$$OK\,or\,FAIL \tag{A3\#2}$$

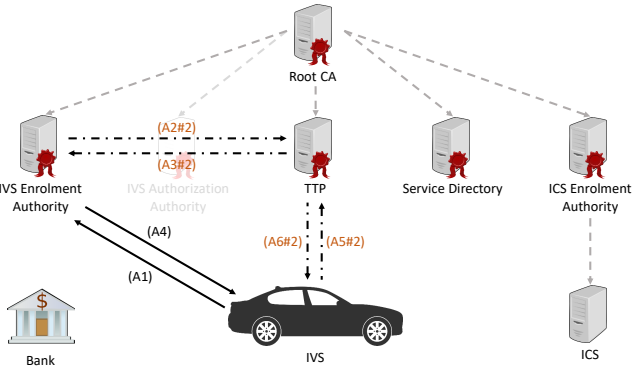


Figure 4.1.: Procedural steps for requesting EC and AC

If the IVS EA aims later on at revoking the IVS, it needs to ask the TTP to remove the unique identifier from the accumulator. It is also possible to store one identifier for each attribute in order to allow revocation of single attributes.

In the next step the IVS EA issues, for both versions, the EC and AC to the IVS. The AC contains all attributes of the IVS. It also includes a hash of the EC. This ensures that the AC is also no longer valid, when the EC is revoked and thus replay attacks are excluded. We describe how the protection works later in Section 4.3.3. When the second version is applied, the AC contains also the unique identifier stored in the accumulator to support revocation.

After issuance, the IVS EA sends for both versions the EC and AC encrypted to the IVS (Step A4). After reception, the IVS decrypts, validates, and stores them.

$$Enc_{EC_{EA}}(EC_{EA}, AC_{EA}) \tag{A4}$$

For the second version the IVS now encrypts the hash of its EC for the IVS EA. Then, it creates a token containing the attributes of the AC received from the IVS EA, a witness to prove that it is contained in the accumulator, and a proof for the encrypted hash of the EC. This token is then encrypted together with the encrypted hash and a certificate request for the TTP. Afterwards, it is sent as Step A5#2 to the TTP. The witness is calculated from accumulator information and needs to be updated each time the accumulator is modified.

$$Enc_{TTP}(cert.request, token, Enc_{IVS_{EA}}(H(EC_{EA}))) \tag{A5\#2}$$

On reception the TTP verifies the token. Furthermore, it saves the encrypted hash. This hash can later on be exploited to reveal the identity of the IVS. Afterwards, it

issues a new EC and AC containing the same attributes as the ones from the IVS EA and, in addition, the hash of the new issued EC to prevent replay attacks. Then it sends them as Step A6#2 back to the IVS.

$$Enc_{EC_{TTP}}(EC_{TTP}, AC_{TTP}) \tag{A6\#2}$$

In general, the verifier of a proof can identify the issuer of the credential. Each OEM will probably operate its own IVS EA. Therefore, a verifier has the possibility to obtain the OEM of an IVS, whereas the IVS might be aiming at hiding this information. If there exists one central TTP which issues the credentials this is no longer possible. Accordingly, the verifier learns only that the credentials are issued by the TTP for all IVSs when the second version is applied.

For simplicity in the remainder EC means EC_{EA} when the first and EC_{TTP} when the second version is applied. Accordingly, when we mention AC, we mean AC_{EA} when the first and AC_{TTP} when the second version is exploited.

4.3.2. Money Request

After the IVS received its EC and AC, it may request digital money from the bank as illustrated in Figure 4.2. This can be done multiple times. Blind signatures might be applied as digital money [Cha82]. If blind signatures are exploited, the IVS generates random values, blinds them, and sends the blinded values together with its authorization to the bank in order to request money (Step B1). The bank then blindly signs the values, debits the bank account of the driver, and returns the blind signed values back to the IVS (Step B2).

$$blinded\_values, auth\_info \tag{B1}$$

$$digital\_money \tag{B2}$$

The IVS now possesses digital money from the bank, which may be used to pay for ATs and thus application usage.

4.3.3. Attribute-Based Authorization Ticket Request

After receiving an EC, AC, and digital money, the IVS is now able to request attribute-based ATs. These ATs are requested from the IVS AA as illustrated in Figure 4.3. The single steps are detailed in the sequel.

In advance of requesting ATs, the IVS queries the SD for suitable applications (Step C1). The SD then replies with a list of ICSs providing applications satisfying the criteria in the query (Step C2).

$$Query \tag{C1}$$

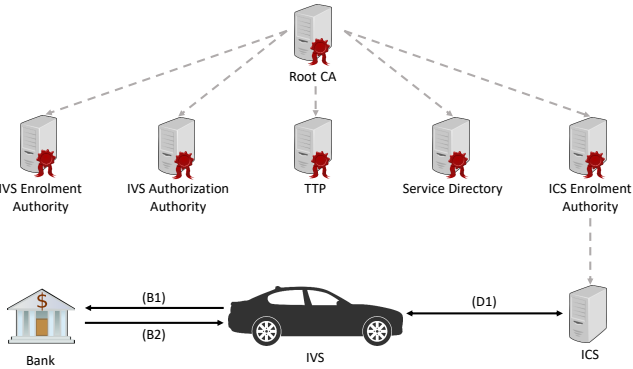


Figure 4.2.: Procedural steps for requesting money and utilization of ATs

$$Possible\,\text{ICS}s \tag{C2}$$

The IVS then selects a suitable ICS and checks if the attributes necessary for the application are present (Step C3).

Now, the IVS first encrypts the hash of its EC for the IVS EA if the first or for the TTP when the second version is applied. Then, it derives a token containing the necessary attributes for the ICS and a proof for the encrypted hash from its AC. When the second version is applied, the token also contains a witness for the revocation information embedded in the AC. Afterwards, it creates the key pairs to certify. Subsequently, it signs the name of the application and the public keys with its EC. This signature is then, together with the EC, encrypted for the IVS EA or TTP, respectively. Finally, the IVS sends the application name, the encrypted parts, the token, the PKs, and the necessary money as given in Step C4 to the IVS AA. The encrypted hash binds the request to a specific EC. The other encrypted part is applied like in the previous work to detect replay attacks [BSS+11]. In the first version, it serves also as a proof that the IVS is not revoked.

$$\begin{aligned} &Enc_{IVS\_EA/TTP}(Sig_{EC}(app\_name, PKs), EC), app\_name, PKs, \\ &digital\_money, token, Enc_{IVS\_EA/TTP}(H(EC)) \end{aligned} \tag{C4}$$

For the next three steps the two versions differ. Therefore, we will describe them independent of each other.

When the first version is applied, the IVS AA sends the encrypted parts together with a hash over the application name and the public keys to the IVS EA (Step C5#1).

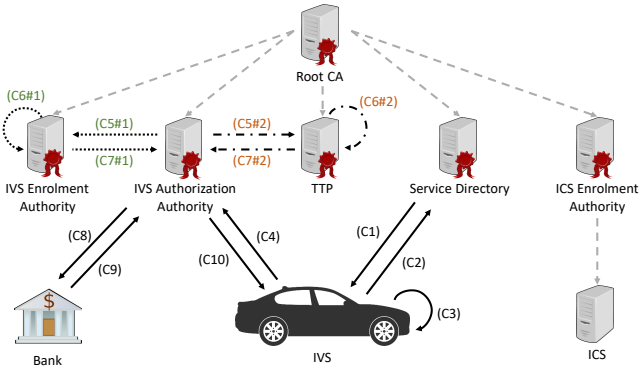


Figure 4.3.: Procedural steps for requesting ATs

$$H(app\_name, PKs), Enc_{IVS\_EA}(hash), Enc_{IVS\_EA}(Sig_{EC}(app\_name, PKs), EC) \quad \text{(C5\#1)}$$

The IVS EA decrypts everything and checks in Step C6#1 as depicted in Figure 4.3 if the EC and the signature are valid and if the hash fits this EC. This ensures that the IVS is not revoked and the applied AC corresponds to the applied EC. Afterwards, the IVS EA returns the result of this validation to the IVS AA (Step C7#1).

$$OK\ or\ FAIL \quad \text{(C7\#1)}$$

In case that the second version is applied, the IVS AA does not ask the IVS EA to validate the request because it cannot determine the IVS EA being responsible for the IVS. Instead, it sends the parts encrypted for the TTP together with a hash over the application name and the public keys to the TTP (Step C5#2).

$$H(app\_name, PKs), Enc_{TTP}(hash), Enc_{TTP}(Sig_{EC}(app\_name, PKs), EC) \quad \text{(C5\#2)}$$

Like the IVS EA, the TTP checks in Step C6#2 whether the EC, signature, and hash are valid. In this version this ensures that the AC is applied with the corresponding EC, but not that the IVS is not revoked. After a successful check, the TTP replies with the current state of the accumulator to the IVS AA (Step C7#2). With this state it is possible for the IVS AA to check if the IVS is revoked.

$$accumulator\ state\ or\ FAIL \quad \text{(C7\#2)}$$

On reception, the IVS AA validates for both versions the token in order to check the attributes of the IVS. When the second version is applied, it also checks the revocation status.

If the validation was successful and the IVS has to pay for the application usage, the IVS AA forwards for both versions the digital money received from the IVS to the bank (Step C8). There the money is transferred to the account of the IVS AA. Because blind signatures are applied for the digital money, the bank is not able to determine the IVS spending this money. Afterwards, the bank returns whether the payment was successful or not (Step C9). On a regular basis the IVS AA transfers part of the money to the ICSs of the issued ATs.

$$digital\_money \tag{C8}$$

$$OK\ or\ FAIL \tag{C9}$$

If all previous steps were executed successfully, the IVS AA finally issues the attribute-based ATs for the application to the IVS (Step C10). Otherwise an error is returned.

$$ATs_{app\_name}\ or\ Error \tag{C10}$$

In order to determine for which application an AT is valid, the identifier of the corresponding application is included. Therefore, the application can check, whether the AT is eligible to use the application. For the remainder of this thesis we assume the AID as part of the ATs is exploited for this purpose as for safety communication [ETS13].

The outlined system supports two different kinds of billing — per request or per time period. If the IVS has to pay per request, each AT may be applied for one request only. When the IVS exploits an AT to access an application, the AT is revoked immediately at the ICS to prevent further usage. The IVS gets as many ATs as it has paid for. When a per time period billing is applied, the IVS gets ATs, which are only valid for the time period the IVS paid for.

4.3.4. Application Usage

As soon as the IVS successfully requested ATs from the IVS AA, it is able to start using the application with the issued ATs as illustrated by Step D1 in Figure 4.2. The obtained ATs might also be applied in application-specific communication with other IVSs.

The system supports the case where each ICS operates its own IVS AA. However, this affects the privacy of the IVS, because the ICS is then in the position to link multiple ATs to the same IVS.

4.3.5. Revocation

The outlined system supports different kinds of revocation. It is possible to revoke a whole IVS from participating in the system as well as single ATs, the whole AC or single attributes. In the sequel these situations are discussed.

ITS Vehicle Station

In both versions the IVS EA has to revoke the IVS. In the first version the IVS EA revokes the EC of the IVS in its local database. There is no difference in comparison to the revocation of an IVS in previous work. When applying the second version, however, the IVS EA has to trigger the TTP to remove the identifier of the IVS from the accumulator.

Therefore, the revocation of a whole IVS is supported by both versions. If revoked, then the IVS is no longer able to obtain new ATs from the IVS AA. Each time ATs are requested the validity of the EC is checked in the first and the state of the accumulator in the second version.

Authorization Tickets

When ATs are revoked, the same mechanism is applied for both versions. If an ICS aims at revoking a single AT for its own application, the ICS may revoke it directly by locally marking it as invalid. An AT is only valid for one application. Therefore, it can be no longer applied to exploit the application. If ATs are also applied in application specific C2C communication the ICS may, in addition, distribute Certificate Revocation Lists (CRLs) to its subscribers to notify them about the revocation. If ATs were issued by accident, the IVS AA can report the affected ATs to the ICS. There they can be revoked.

Anonymous Credential

The revocation of the AC can be done by revoking the whole IVS. However, the IVS is then also no longer able to obtain ATs for safety communication. For the first version the IVS EA can only mark the AC of the IVS as invalid. Then it can reject all requests for ATs which are not intended for safety communication. When the second version is applied, the IVS EA can trigger the TTP to remove the identifier of the IVS from its accumulator. Then the IVS is no longer able to obtain any new attribute-based ATs too. However, it can still get new ATs for safety communication, because the TTP is not involved in this case.

Attribute

To revoke a single attribute of an IVS, it is necessary to have some kind of misbehavior detection. It is necessary to detect which attributes are no longer valid. There is a need for a central entity which collects misbehavior reports containing the AT and in which manner the IVS misbehaved. These reports might be created by the ICSs and trigger the revocation process. Furthermore, this information might be exploited to trigger a repair at the workshop.

For the first version the identity of the holder of an AT can be revealed by a cooperation of the IVS EA and IVS AA, like for safety communication. The IVS AA knows the AT, the IVS EA the corresponding identity and both know the same hash from the request. After the identity is revealed, the IVS EA can mark the AC of the IVS as invalid and record the misbehaving attribute. The next time the IVS tries to request a new AT for an application, the IVS EA detects that the AC is no longer valid and triggers the IVS to request a new AC. The new AC then has less or other attributes depending on the misbehavior of the IVS.

When the second version is applied the TTP is in addition to the IVS EA and IVS AA involved in revealing the identity of the IVS. The IVS AA knows the AT of the misbehaving IVS. The TTP can link it with the hash of the request to the EC it issued. Furthermore, it is in possession of the encrypted hash of the corresponding EC issued by the IVS EA. Therefore, this encrypted hash is then forwarded to and encrypted by the IVS EA in order to reveal the identity of the AT holder. In addition, the IVS EA marks the misbehaving attribute in its database. If the accumulator contains a revocation information of this attribute, it is removed by the IVS EA. If not, the identifier for the whole AC is removed from the accumulator. Whenever a proof verification for this IVS fails, it might obtain a new AC with changed attributes.

4.4. Implementation

A prototype implementation supporting both versions of the advocated scheme was done as part of [Azi15]. As the foundation of the anonymous credential subsystem we exploited *idemix* [CVH02]. To issue ATs and ECs, we used the Pilot-PKI of the C2C-CC. All software modules are written in Java. Web services are employed in order to support the exchange of data between the different components. The components of the prototype implementation and their interactions are illustrated in Figure 4.4.

A Graphical User Interface (GUI) called OEMview was created to register IVSs with their attributes at the IVS EA. The IVS EA stores this information in a local SQLite database.

Another GUI denoted as IVSview was created in order to evaluate the interactions

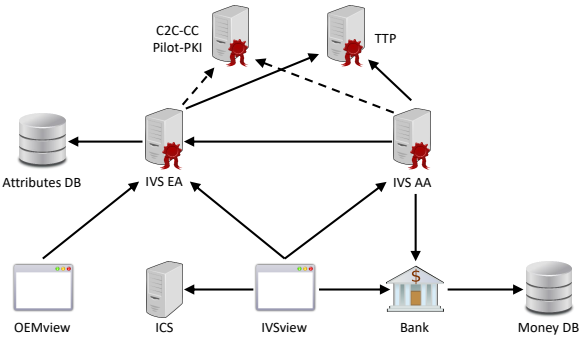


Figure 4.4.: Components of and interactions within the prototype implementation

of the IVS with other components. One part of IVSview allows the enrolment of the IVS and therefore to get the EC and AC from the IVS EA. The IVS EA also registers the IVS at the Pilot-PKI and stores the revocation information in its local database. The IVSview also allows to request digital money from the bank. To support different money values, the bank applies a different key for each possible value. The bank also has a local SQLite database, where all exploited bank notes are stored to prevent double spending.

In a different part of the IVSview, it is possible to request ATs. Prior to creating the request, it is necessary to select an application and money amounts. In the prototype implementation the IVSview and IVS AA both have a local copy of the SD in order to select an application and to verify the necessary permissions. When executing the request, the user is able to decide which version of the scheme shall be applied. The IVSview contacts the AA for the ATs, which communicates, depending on the selected scheme version, with either the IVS EA or TTP. As attributes of the IVS we applied the brand, production date, and sensors for rain, Bluetooth, and its position. We defined 20 more sensor attributes in order to evaluate the scheme for applications with different numbers of required attributes. We created various applications ranging from some requiring only one attribute to some requiring up to 20. We did not implement the issuance of the second EC and AC by the TTP. However, the scheme would operate in the same way, just with another AC and EC.

4.5. Evaluation

For evaluation purposes we first compare the properties of the two proposed versions and we then evaluate their performance.

4.5.1. Comparison

Both versions enable an IVS to authenticate at an ICS with their previously obtained attribute-based ATs. In order to obtain these ATs, the IVS needs to prove to the IVS AA that it possesses the necessary attributes for this application. However, only the attributes necessary to exploit the application are revealed. In addition, different proofs cannot be linked to the same IVS. Therefore, neither the IVS AA nor the ICS are able to determine the attributes not necessary for the targeted application. Furthermore, no entity in the system is able to link ATs to an IVS. The IVS EA respective TTP knows the identity of a requesting IVS but not the issued ATs. On the other hand, the IVS AA and ICS are aware of the issued ATs, but are not able to identify the IVS. Therefore, none of the entities knows the identity of the IVS and issued ATs and is able to link them. Only when the IVS EA and IVS AA cooperate it is possible to reveal the identity of an IVS. If the second version is applied the TTP is in addition necessary to reveal the identity.

In case of the first version, the IVS EA validates the request of the IVS. It is assumed that each OEM will operate its own IVS EA. Therefore, it is possible for an IVS AA to determine the OEM which manufactured the IVS. The second version hides this property. There, each IVS requests a second Anonymous Credential with the same content from a central TTP. This AC is then used to prove the attributes. Consequently, the IVS AA contacts the TTP to validate the request. Because the TTP is responsible for IVSs from different OEMs, it is not possible to determine the OEM of the requesting IVS. A drawback of this version is that each IVS needs a second EC and AC. Furthermore, it complicates the revocation, because it cannot be done locally by the IVS EA anymore. The IVS EA needs to contact the TTP when an IVS is revoked. Furthermore, the witness of the IVS needs to be updated periodically. When revealing the identity of an AT holder the TTP has to be involved too.

In the first version the AC needs no special protection. An attacker obtaining it is not able to successfully request ATs. During every request the IVS EA checks if the requester is in possession of the corresponding EC. In the second version the AC contains the revocation information. An attacker may exploit this information to mount an attack aimed to revoke the IVS. Therefore, the AC should be saved in secure storage, if the second version is used. Furthermore, also the EC obtained from the TTP needs to be stored there.

This comparison illustrates that the second version safely protects the IVS AA

Table 4.1.: Comparison of the scheme versions

Property	Version 1	Version 2
Hides unnecessary attributes	✓	✓
Hides the identity	✓	✓
Hides the OEM	-	✓
No revocation updates necessary	✓	-
Keeps System complexity low	✓	-
Introduces no new entities to trust	✓	-
Minimizes necessary secure storage	✓	-

from obtaining the OEM of the IVS. However, this property comes with a more complex system, a new entity to trust, periodic updates of the revocation witness, and the need for more secure storage. Table 4.1 summarizes the comparison results. It is not necessary that all OEMs employ the same version, they can easily coexist side by side. Even IVSs from the same OEM might exploit different versions.

4.5.2. Performance

In order to evaluate the overall performance of the proposed scheme, we used the following setup. The performance measurements of the components running on the IVS were taken on a NEXCOM VTC6200 CarPC with an Intel Atom D510 Dual Core CPU operated at 1.6 GHz and featuring 2 GB of memory. The tasks of the Pilot-PKI were executed on a server, hosted by the operators. The performance measurements of the remaining entities were taken on a notebook equipped with an Intel Core i5-3360M running at 2.8 GHz clock frequency and featuring 8 GB of memory. An overview of the measured times is given in Table 4.2 and depicted in Figure 4.5. We discuss the measured execution times in more detail as follows.

Enrolment Process

This process is done once for each IVS and includes the registration of the IVS at the IVS EA and the request of its EC and AC. It takes 279 ms to register an IVS with its properties at the IVS EA. This value includes 120 ms to register it at the Pilot-PKI. The registration operation is not time critical. It might be done prior to production.

The complete request to obtain the EC and AC takes 592 ms. This value includes the request of the EC from the Pilot-PKI (114 ms), the issuance of the AC (314 ms), and the addition of the identifier for revocation to the accumulator (22 ms). However, this operation is not time critical at all, since it is being performed during the assembly of the IVS. Therefore, there is plenty of time to execute.

Table 4.2.: Execution times of scheme operations

Operation	Time
Registration (Complete)	279 ms
Registration (Pilot-PKI)	120 ms
Request EC and AC	592 ms
Issue EC (Pilot-PKI)	114 ms
Issue AC	314 ms
Add identifier to accumulator (only version 2)	22 ms
Request money	49 ms
Verify EC at EA (only version 1)	55 ms
Verify request at TTP (only version 2)	56 ms
Cash money	22 ms
Issue AT (Pilot-PKI)	340 ms

Money Request

This request needs to be executed each time the IVS has not enough money available in order to exploit a certain application. Therefore, it should be executed it in a reasonable time. In the prototype implementation it took on average 49 ms, which is acceptable, but it certainly may be improved by an optimized final implementation.

Attribute-Based Authorization Ticket Request

An IVS needs to request new ATs from the IVS AA each time it aims at exploiting a new application or it owns no valid ATs for a booked application anymore.

Figure 4.5 illustrates for version 1 the total time necessary to create a proof for the requested attributes and to request new ATs from the IVS AA depending on the number of attributes to prove. Furthermore, it also shows the time necessary to create the proof at the IVS and to validate it at the IVS AA.

The figure clearly indicates that all three times increase linearly with the number of attributes included in the proof. The total time increases from 2.1 seconds, when only one attribute is required, over 8.5 seconds, when 10 attributes are required, up to 15.6 seconds, in case that 20 attributes are needed by the application.

The figure shows furthermore that most of the time is consumed while creating the proof of attributes. For one attribute it is 70 % (1.5 seconds), for 10 attributes it rises to 87 % (7.7 seconds), and for 20 attributes finally to 90 % (14 seconds) of the complete time. The time to validate the proof increases from only 0.1 second for one attribute up to 1 second for 20 attributes.

The execution time to validate the request differs only by a few milliseconds for

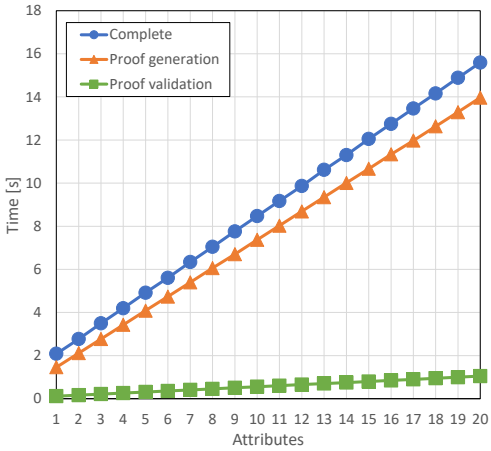


Figure 4.5.: Execution time of an AT request as a function of the number of attributes for the first version

the two scheme versions. It takes 56 ms to validate the request at the TPP and to get the current state from the accumulator. For version 1, it takes 55 ms to validate the EC at the IVS EA. However, when the witness for the revocation information of the AC is included as part of the proof generation in the second version, around additional 570 ms are necessary. The time to validate the proof increases by approximately 50 ms. In order to get the execution time for the proof generation, proof validation, and complete execution for version 2, these times have to be added to the numbers displayed in Figure 4.5. The final time further includes the time to cash the money at the bank (22 ms) and the request of ATs from the Pilot-PKI (340 ms).

If the accumulator applied in the second version also contains revocation information for single attributes, the execution time increases further. For each additional identifier in the accumulator around 570 ms are additionally required for generation and around 50 ms for validation, respectively. Figure 4.6 illustrate the introduced overhead for a given number of revocable attributes included in the proof. These time values need to be added to the values displayed in Figure 4.5 in order to get the total execution time for version 2 of the scheme. The revocation process needs only a few milliseconds for both versions.

Obviously, it takes several seconds to generate the proof. An IVS requesting information from an ICS in general is not likely to wait for such a long time. For this reason, we apply the anonymous credentials system and the proof generation only when new ATs need to be obtained. Therefore, an IVS only has to generate a

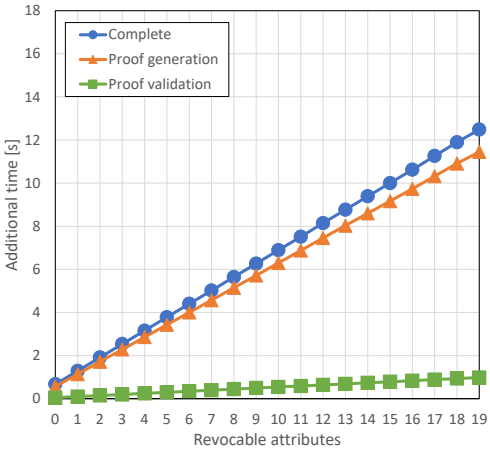


Figure 4.6.: Necessary additional time to execute the second version as a function of the included revocable attributes

proof whenever it aims in exploiting an additional application or does not hold valid ATs for the ICS anymore. For the communication with the ICS, it applies the ATs. It takes just 6 ms to generate a valid signature on the same hardware with a valid AT [BBH15]. Furthermore, it is expected that production IVSs will feature an adequate hardware acceleration aimed to significantly reduce the signature generation time.

In case that a new application is envisaged, the process of proving the attributes might be done in parallel to the download and installation of the corresponding application. Therefore, no additional delay will be introduced for the IVS. In case that the IVS needs to renew its ATs, the operation to request them can be executed in background prior to an expiration of the old ones. Again, no waiting time will be introduced. This is however not possible, if the proof needs to be generated for each single request to the ICS. Last, but not least, please note that our prototypical implementation was not optimized for low execution time. Thus production grade implementations can be expected to show significantly decreased execution time.

4.6. Summary

In this chapter we introduced and discussed a novel approach aimed to allow ICSs to specify attributes an IVS needs in order to use an envisaged application. An IVS

can prove to the IVS AA by means of an AC that it owns the necessary attributes. The mandatory proof is done without revealing any attributes not necessary for this purpose. The IVS AA then issues attribute-based ATs for this application to the IVS. Afterwards, the IVS can apply these ATs to prove to the ICS that it is eligible for the envisaged application. Again, this is done without revealing its real identity. The ICS only learns the attributes necessary to obtain the ATs. We assume a central billing entity, which supports both request and time based billing. If digital money is exploited, no entity can link the spent money to a certain IVS. Since all applications can exploit this billing service, ICSs do not need to create their own procedures. Therefore, this scheme protects the privacy of the IVSs.

Each AT allows only to exploit one application. Subsequently, it is not possible to get information about the other application an IVS uses from an AT. Therefore, the scheme also protects the privacy of association of the IVS. All communication between the backend entities can be easily secured by e. g. TLS. The IVS knows the identity of all entities it is exchanging messages with. Therefore, it is also possible to secure this communication so that the privacy of communication is preserved for the IVS.

We presented two different versions of the advocated scheme. Both versions support revocation on different levels. Furthermore, both preserve the privacy of the IVSs against the different entities. As the main result, no entity can link an AT to the identity of an IVS. We compared the fundamental properties of these versions. The first one leaks the OEM of the participating IVS. The second one overcomes this drawback but features a higher complexity, needs periodic updates of its revocation information, introduces a new entity to trust, and needs more secure storage. However, both versions can be exploited simultaneously by different IVSs depending on their specific requirements.

With the help of a prototype implementation we showed that the proposed versions differ in terms of the execution time. However, the time delay introduced by the AC system in both versions does not affect the usage of the application. The time consuming operations may well be done prior to the requests to the ICS. Hence, the proposed approach hides the attributes not necessary for a specific application and consequently protects the privacy of the participating IVSs while not interfering with the application usage.

The scheme proposed in this chapter generalizes the PKI applied for safety communication in VANETs to non-safety communication. It integrates the proof of attributes to exploit a certain application while maintaining the privacy of the vehicles. Furthermore, it integrates billing in the existing scheme. All this is applied without a noticeable delay for the driver.

Open communication architectures like CONVERGE [CON] might exploit the proposed attribute based authentication scheme. This can increase the user accep-

tance of the platform. Furthermore, OEMs can integrate the scheme in their application market places in order to advertise its privacy protection. Additionally, decentralized crypto currencies like Bitcoin [Nak08] might be integrated. Hence, the central function of the bank can be removed. Therefore, this might reduce the overall system complexity and further enhances the privacy of its users.

5 | Anonymous Data Exchange

In the previous chapter we presented a scheme that allows an IVS to obtain ATs which can be exploited to authenticate itself at an application which requires certain attributes. In this chapter we propose a protocol which applies these pseudonyms in order to establish an anonymous authenticated confidential data exchange between two IVS running the same application. This kind of protocol is necessary if, e. g., two IVSs aim at exchanging data which is confidential to subscribers of a specific application. In such a case the data should not be readable by other IVSs.

This chapter is based on the publications [BH15c], [BH15d], and [BBH15]. In addition to the papers we describe and evaluate how the transmitted data can be compressed to save bandwidth, how the overall execution time can be reduced, and how real payload can be exchanged by means of the protocol. Furthermore, we extended the description of the implementation and evaluation results.

5.1. Motivation

For safety reasons, IVSs in VANETs use broadcast over DSRC to inform other IVSs in communication range about their current status, including position, heading and speed. All safety messages are digitally signed to prove the integrity of the message and eligibility of the sender. They are not encrypted, since every participant shall receive and interpret these messages with as less processing time as possible. Besides this safety relevant communication, there are other applications where the data exchanged between IVSs is confidential and therefore needs to be encrypted.

Consider for example an application which offers software updates or up to date map data to IVSs. In order to save bandwidth over mobile networks like LTE the operator of the application could decide to distribute the updates only to a subset of IVSs. These IVSs could then distribute the data via epidemic distribution [LHH08] over DSRC to other IVSs which did not yet receive the update.

Another application could for example collect by means of its built-in sensor data like traffic signs or potholes. IVSs thus record the sensor data and report it to the ap-

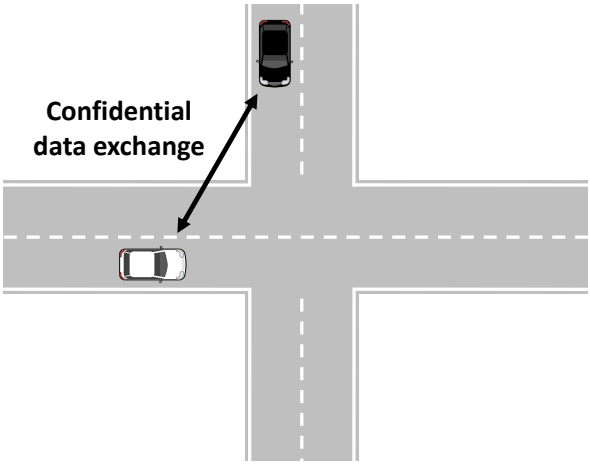


Figure 5.1.: Principle of the protocol

plication server, which offers the data to other applications or local administrations. The ICS may aim at allowing only IVSs with certain sensors to participate in the application. This application sounds simple, but it introduces a huge privacy risk. The data sent to the ICS contains privacy related data like the location of the sensor data and a coarse time, when it was detected. If all IVSs would simply sent their detected data to the ICS, then this entity is able to create movement profiles of the IVSs and therefore hurt their privacy of location and space. Even visited sensitive locations like hospital, home, or work place may be extracted from this data. This information can then be exploited to identify a person and therefore violate their privacy of the identity. In order to preserve the privacy of its users and to advertise the privacy considerations, the ICS decides to implement a privacy preserving mechanism. This mechanism requires that the IVSs exchange their collected data between each other prior uploading. When using this mechanism, the IVSs report sensor data locations detected by other IVSs too. Therefore, the ICS can no longer determine where a specific IVSs was driving at the given point in time and can no longer create movement profiles of these IVSs. The ICS could decide to exploit DSRC to exchange the data between the IVSs, because it is free of charge. In addition, the ICS enforces an encrypted data exchange policy, so no attacker can record and sell the collected data next to the ICS and hurt the privacy of communication of the IVS. The exchange of the encrypted data shall also be privacy preserving. We analyze this privacy preserving mechanism later in Chapter 6. The basic principle of the protocol is illustrated in Figure 5.1.

There are three problems to be considered, when confidential data is being exchanged between IVSs:

(1) How does an IVS ensure whether the other IVS is eligible to receive confidential data?

(2) How do IVSs exchange the key for encrypting the communication data?

(3) How can the first two problems be solved while preserving the privacy of the IVSs?

For safety-related communications in VANETs each IVS applies a pseudonym to sign all outgoing messages. To ensure the privacy of the IVS the pseudonym is changed on a regular basis. This is done in a way that it is not possible to link two pseudonyms to the same IVS. Of course, an IVS might simply use an own set of pseudonyms for each application by encoding the application into the pseudonym as outlined in Chapter 4, and exploit well-known key agreement protocols like ECIES as standardized for VANETs in [IEE13b] to authenticate against each other and finally to agree on an encryption key. However, the IVS then needs to change its pseudonym for the application at hand at the same time as the one intended for safety-related communication to prevent linking of pseudonyms. This introduces a cost overhead both for additional secure storage for the private keys and for the data transmission aimed to obtain new pseudonyms. Therefore, we propose in the sequel a novel protocol that solves all three outlined problems and at the same time reduces the number of necessary pseudonyms in comparison to exploiting ECIES only.

To do so we combine ring signatures [RST01] with the ECIES scheme [IEE13b]. In order to address the receiving IVS, we also had to define a message structure the actual protocol is embedded into.

We evaluated the resulting protocol with respect to the privacy of the IVS. To do so we analyzed the information different types of attackers might gain and simulated the protocol in a VANET simulator. Going beyond simulation, we furthermore implemented and empirically evaluated this protocol by means of real vehicles. We analyzed the message size and further evaluated the implementation regarding the complete execution time of the protocol, the time necessary to execute the individual protocol steps, and the execution times of the different protocol phases. We did this for different ring signature sizes, payload sizes, and vehicle velocities. We also evaluated the underlying implementation regarding fault tolerance.

5.2. Protocol

An anonymous authenticated key agreement protocol allows two parties, who are members of the same group, to establish a confidential communication without leak-

ing their identity. To achieve this goal, both parties have to agree on a session key to encrypt the exchanged data. The identity of the other party is unknown at the beginning of the protocol and both parties are not willing to expose for privacy reasons their application-specific identity to anyone. In addition, it shall be possible to revoke access for single parties and only members of the same group shall be able to agree on the session key. The protocol shall fail, if one party is not a member of the group. Not eligible parties shall gain as few information as possible about the other party. We only consider single-hop connections, because multi-hop connections are difficult to maintain in VANETs due to frequent topology changes. An IVS which does not belong to the group shall gain as little information as possible by eavesdropping the exchanged messages.

In the remainder of this section we first describe how the applied cryptographic mechanisms of the anonymous authenticated key agreement protocol work together. Then, the applied notation is explained before the single steps of the protocol are detailed. Finally, the format of the GeoNetworking messages employed to embed the protocol and address the destination IVS is described.

5.2.1. Applied Cryptographic Mechanisms

During execution of the protocol two IVSs agree on a symmetric encryption key to exchange confidential data by applying ECIES as standardized in [IEE13b]. As only IVSs exploiting a specific application shall be able to agree on the symmetric encryption key, ECIES is combined with ring signatures [RST01]. This is done by creating a ring signature with application-specific pseudonyms over the ECIES parameters. Thus, it is ensured that both IVSs are authorized to use the application. If one of the IVSs is not authorized for the application, it is not able to create a valid ring signature over the ECIES parameters.

This generic approach has the advantage that IVSs can apply the safety identities already known to each other for ECIES and hide the application-specific identity by means of ring signatures. When applying ring signatures, it is not possible to determine the pseudonym of the signature creator. A ring signature is, as described in Chapter 2, created by applying n different pseudonyms, whereas for one the signer must be in possession of the corresponding private key. For all other pseudonyms it is sufficient to only know the public keys. For a verifier it is not possible to determine the pseudonym of the signer from the other applied pseudonyms. The verifier can determine that the identity of the signer is one of the n applied pseudonyms, but not which. Therefore, pseudonyms can be reused in different ring signatures without being linked to the same IVSs. As a consequence, less application-specific pseudonyms are necessary in compare to the pseudonyms applied for safety communication.

When applying the same application-specific pseudonym multiple times it is nec-

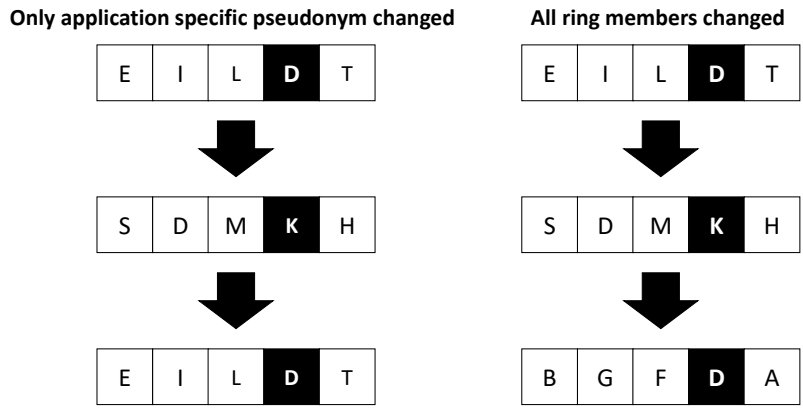


Figure 5.2.: Examples for changing ring members

essary to apply different ring members. Otherwise, the whole ring signature could be linked to the same IVS. In order to create a ring signature, the IVS needs to create a pool of valid pseudonyms. Each time a new ring is created, the IVS selects randomly $n-1$ pseudonyms out of this pool. The collection could be achieved by storing the pseudonyms received from other IVSs while executing the protocol. When an IVS applies the same application-specific pseudonym in different rings with different safety-related pseudonyms, an attacker is not able to link the two safety-related pseudonyms applied to the same application-specific pseudonym. This is illustrated in Figure 5.2. On the left side of the figure the upper ring with the own application-specific pseudonym D is reused in the lower ring without changing the other ring members. Therefore, an attacker can easily link the two rings to the same IVS. On the right side of the figure the own application-specific pseudonym D is also reused in the lower ring. However, all other ring members have been changed. Therefore, an attacker is not able to link the two rings containing the pseudonym D to the same IVS. It could be also another IVS including the earlier recorded pseudonym D in its ring.

We apply the ring signature scheme based on ECs as proposed in [LLZ+07]. We favor this scheme, since ECs provide the same security level with a much shorter key and signature length compared to other schemes like Rivest, Shamir, and Adleman (RSA). Furthermore, ECC is also applied in VANETs to secure the safety-related communication. Therefore, IVSs in VANETs will feature dedicated hardware to speed up EC calculations.

We propose two versions of the protocol. When applying the basic version of the

protocol, only the ring signature is encrypted with the key exchanged by ECIES, while the second version in addition encrypts the applied application-specific pseudonyms and ring signature values necessary to validate it. We denote these versions the plain and encrypted version of the protocol, respectively.

5.2.2. Notation

The following notation is applied to describe the protocol. *SAM* denotes the SAM according to [ETS10a]. V, C and T are the ECIES parameters as standardized for VANETs in [IEE13b]. V is the public key of the sender, the parameter C is the symmetric AES key K encrypted by ECIES, while T denotes the authentication tag of ECIES. $Cert_{Xn}$ are the n possible application-specific pseudonyms of entity X, where one is the application-specific pseudonym of the IVS and the others are the collected pseudonyms of other IVSs. For each possible signer of the ring signature, a validation value x_{Xn} is necessary. σ_X denotes the ring signature created by entity X. When Y is encrypted by the symmetric encryption key K, it is written as $E_K(Y)$.

5.2.3. Protocol Steps

The plain version of the protocol works as follows:

	A $\rightarrow$ *:	SAM
(1)	B $\rightarrow$ A:	$V, C, T, Cert_{B1}, ..., Cert_{Bn}, x_{B1}, ..., x_{Bn}, E_K(\sigma_B)$
(2)	A $\rightarrow$ B:	$Cert_{A1}, ..., Cert_{An}, x_{A1}, ..., x_{An}, E_K(\sigma_A)$
(3)	B $\rightarrow$ A:	$E_K(payload)$
(4)	A $\rightarrow$ B:	$E_K(payload)$
(5)	B $\rightarrow$ A:	ACK

Alice (A) periodically sends a SAM via broadcast to all IVSs in communication range to indicate that she uses an application which takes advantage of the advocated protocol.

Assuming Bob (B) who is in communication range and also utilizes this application receives a SAM from Alice. Then he generates a random AES key K and calculates the ECIES parameters V, C, and T. Next, he selects $n-1$ pseudonyms from his collected pool of application-specific pseudonyms and his current for this application in order to calculate his ring signature σ_B over V, C, and T. Then, he encrypts σ_B with the symmetric AES key K. Finally, he sends the resulting ciphertext with the ECIES parameters V, C, and T, the applied application-specific pseudonyms, and the additional ring signature values as Step 1 to Alice.

After reception Alice decrypts the AES key K according to the ECIES scheme and applies it to decrypt the ring signature σ_B. Afterwards, Alice validates the ring signa-

ture. On successful validation, she selects $n-1$ collected and her current application-specific pseudonym for this application. With this set of pseudonyms, she creates a ring signature σ_A over the ECIES parameters V, C and T. Afterwards, she encrypts the resulting signature with K. Subsequently, she sends the ring signature and everything necessary to validate it to Bob (Step 2).

On reception, Bob first decrypts and validates the ring signature σ_A. After successful validation, Alice and Bob are certain that the other party is authorized to employ the application. In addition, both are in possession of the same encryption key K still without knowing the application-specific identity of the other party. Therefore, Bob now sends his confidential payload encrypted with the symmetric encryption key K as Step 3 to Alice. In the following Step 4 Alice responds with her confidential payload, also encrypted with K.

When all confidential payload is exchanged, the protocol is terminated by Step 5 of the protocol which consists of an acknowledgment sent from Bob to Alice.

Compared to the plain version of the protocol the encrypted version encrypts not only the ring signature σ_X, but also the pseudonyms applied to create the ring signature and the additional ring signature values. Therefore, Steps 1 and 2 of the protocol look like the following for the encrypted version of the protocol.

$(1') \text{ B} \rightarrow \text{A}: \quad V, C, T, E_K(Cert_{B1}, ..., Cert_{Bn}, x_{B1}, ..., x_{Bn}, \sigma_A)$
$(2') \text{ A} \rightarrow \text{B}: \quad E_K(Cert_{A1}, ..., Cert_{An}, x_{A1}, ..., x_{An}, \sigma_B)$

When the application-specific pseudonyms are encrypted a potential attacker gets less information which can be applied to identify the IVS. In Section 5.3 we analyze the capabilities of different attackers to both protocol versions in detail.

Given that the other party and a potential attacker already know the pseudonym used for safety messages, the pseudonym applied to execute the ECIES does not give an attacker any new knowledge. These safety pseudonyms should change on a regular basis and not be reused. Therefore, they cannot be exploited to track anything. The considered goal of an attacker is to determine the identity, e.g., application-specific pseudonym of an IVS since this shall be reused in different ring signatures and may therefore be used to link different safety identities of the IVS. This may be done, whenever an IVS exploits the same application-specific pseudonym twice, but with different pseudonyms for safety communication in VANETs. Then, an attacker could link, as described in Chapter 4 the two safety pseudonyms to one IVS, because they are utilized in combination with the same application-specific pseudonym.

We assume that each IVS can have multiple valid application-specific pseudonyms at a time. So, the IVS is able to change their application-specific pseudonym they use for creating the ring signature regularly with their pseudonyms for safety relevant communication to avoid being tracked by means of the pseudonym not being

changed within the ring signature.

Consider that Alice applies the safety pseudonym $Cert_{S1}$ and application-specific pseudonym $Cert_{A1}$ at the same time. Then, she changes her safety pseudonym to $Cert_{S2}$, while still using the application-specific pseudonym $Cert_{A1}$. An attacker now may link $Cert_{S1}$ and $Cert_{S2}$ because they were exploited with the same application-specific pseudonym.

Multiple pseudonyms confuse an attacker considerably, since each IVS has multiple identities. In addition, these identities could be applied at the same time in ring signatures of different IVSs. The impact of multiple parallel pseudonyms is evaluated in Section 5.4.4.

For the outlined protocol we stick to the same pseudonym format already existing in VANETs to sign safety messages. However, we bind them to a specific application as outlined in Chapter 4. We also apply ECC and ECIES, which is already standardized for safety communication between IVSs. Therefore, this protocol fits very well in the VANET environment.

It is possible to exclude a IVS from successfully executing the protocol by revoking its application-specific pseudonyms. If the pseudonyms of an IVS are revoked it is no longer possible for the IVS to generate a valid ring signature. Other IVSs verifying the signature will detect that the pseudonyms are no longer valid and subsequently abort the protocol execution with this particular IVS. The revocation could be done by, e.g., the distribution of an CRL by an ICS as outlined in Chapter 4.

When applying this protocol, two IVSs are able to prove each other that they are authorized to exploit a specific application without leaking their identity to others. At the same time, they agree on a symmetric encryption key to exchange confidential data.

5.2.4. Message Format

As message format for the outlined protocol we apply GeoNetworking as described in Chapter 2.3.1 and standardized in [ETS10c]. In the sequel we discuss how GeoNetworking can be applied to support the protocol execution.

As a first message to announce the protocol a SAM is sent from the initiating IVS. The message format applied for this message is the same as standardized in [ETS10a] and detailed in Chapter 2.

For the remaining messages two IVS exchange messages between each other. Therefore, unicast messages are most suitable for this communication. A unicast message in GeoNetworking is defined in a GUC as the *Extended Header*. Furthermore, we apply the BTP as *Transport Protocol*. It features in comparison to GeoNetworking-IPv6 (GN6) a smaller overhead and easier handling.

The security profiles for CAM, DENM, and the generic one were not suitable

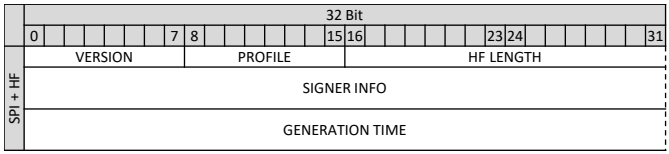


Figure 5.3.: Applied message format

for the outlined protocol. The profiles for CAM and DENM contain a hard coded message type. Furthermore, the generic profile contains like the one for DENMs a generation location, which is not required in our case because the messages are generated by an ICS. We therefore defined an own profile. This profile consists of the signers pseudonym applied for safety communication(*SIGNER INFO*) and the generation time of the message (*GENERATION TIME*) as *Header Fields*. The certificate of the signer is exploited to ensure a trustworthy sender as source of the messages. The generation time of the message is inserted to ensure the freshness of the message and to detect possible replay attacks. The resulting *Header Fields* of the GeoNetworking message with all elements are depicted in Figure 5.3. The overhead introduced by GeoNetworking is also considered in the empirical evaluation of the protocol in Section 5.5.

5.3. A-Priori Assessment

In the a-priori assessment we distinguish between *passive* and *active attackers* to assess how much information they can gain. A passive attacker is only able to listen to and record exchanged messages, while active attackers are also able to replay and send messages under a forged identity. We consider four types of active attackers that differ in their access to pseudonyms. The least powerful attacker has no access to any valid pseudonyms. Another attacker has only access to pseudonyms for safety relevant communication. The third one has only access to application-specific pseudonyms, while the most powerful attacker is an insider and has access to both pseudonym types.

Table 5.1 compares the encrypted and plain version of the protocol regarding the information the different attacker types can yield. We consider the size of the ring and the pseudonyms applied by Alice and Bob as critical. When the attacker is able to obtain the respective information, it is denoted as ✓, otherwise as -.

Regardless of the applied version of the protocol, all attackers are able to calculate the employed ring size from the message size. When the plain version is applied, all attackers can get the pseudonyms utilized by Alice and Bob, since they are trans-

Table 5.1.: Capabilities of different attackers

Attacker	Protocol	Ring size	Alice	Bob
Passive	Plain	✓	✓	✓
	Encrypted	✓	-	-
Active without pseudonyms	Plain	✓	✓	✓
	Encrypted	✓	-	-
Active with safety pseudonyms	Plain	✓	✓	✓
	Encrypted	✓	-	✓
Active with application pseudonyms	Plain	✓	✓	✓
	Encrypted	✓	-	-
Active with safety and application pseudonyms	Plain	✓	✓	✓
	Encrypted	✓	✓	✓

mitted in plain text. Therefore, only the capabilities of the attackers regarding the encrypted version of the protocol are discussed in the sequel.

The passive attacker is not able to get the pseudonyms of Alice and Bob when the encrypted version is used, since they are encrypted and the attacker cannot derive the encryption key K just by listening to the exchanged messages.

Without access to valid pseudonyms, an active attacker is not able to successfully inject any message, since all of them are either signed or encrypted. If the attacker replays the SAM, she cannot encrypt the first or reply a valid second step, since she does not know and cannot calculate the encryption key. If she replays the first step, she is not in the position to decrypt the pseudonyms applied by Alice in Step 2, since she does not know and cannot calculate the encryption key. Therefore, all active attackers are not able to get any information by replaying messages.

An active attacker with access to pseudonyms for safety relevant communication could generate and send the SAM. If she sends the SAM, she cannot replay with the second step, because she has no application-specific pseudonym available in order to generate a valid ring signature. However, she could decrypt the pseudonyms applied by Bob in Step 1 by calculating the encryption key K and therefore can get the pseudonyms employed by Bob.

An active attacker, who has only access to application-specific pseudonyms, cannot generate and send a valid SAM, because it is signed with a pseudonym for safety relevant communication. The same holds for the remaining protocol steps.

If an active attacker has access to both a pseudonym for safety relevant communications and to an application-specific pseudonym, she is in the position to send and to answer to all steps of the protocol with a valid message and therefore gets the pseudonyms applied by Alice and Bob.

This analysis shows that only the most powerful active attacker is able to unveil the identities applied by Alice and Bob when the encrypted version of the protocol is being applied. However, lots of sophisticated work will be necessary to implement this type of attacker in practice, since the private keys of the pseudonyms are in general stored on a Hardware Security Module (HSM) inside the IVS. Of course, if the HSM fails and an attacker is thus able to extract the private keys, she can get both the valid safety and the application-specific pseudonyms and send valid fake messages. However, then the attacker could also extract only the private keys of the safety pseudonyms and link them directly or send valid fake safety messages. In general, it is possible to detect and revoke the affected IVS. Failed HSMs are a general problem in VANETs. Therefore, we will not investigate it in more detail.

5.4. Simulation of Privacy Properties

We built a simulation scenario to assess the outlined protocol regarding its privacy impact in VANETs. We considered different parameters like the strategy on how to select the pseudonyms applied in the ring signature or how many parallel valid pseudonyms an IVS carries. All these parameters influence the privacy of the IVSs in the simulation. We evaluate these parameters regarding a powerful attacker which tries to identify the IVS. For the simulation we exploited VSimRTI [Sch11] with SWANS [Bar04] and SUMO [KEBB12]. For a realistic road network, we imported the streets from Openstreetmap [1].

5.4.1. Scenario

As simulation area we selected the motorway A60 south of Rüsselsheim, Germany. At each junction of the motorway one IRS is placed. This is a realistic placement of IRSs. It is considered to be too expensive to deploy a comprehensive network of IRSs [PHJOZ15]. Therefore, only major intersections and dangerous locations like the junctions of a motorway will be equipped. There it is possible to inform lots of passing IVSs about traffic jams, weather hazards, or alternative routes. In our simulation we assume that these IRSs are under control of an attacker, who is able to record all messages exchanged in communication range.

The IVSs enter the simulation area at two points: One in the east, for the IVSs driving westbound, and one in the west of the map for IVSs driving eastbound. The scenario is illustrated in Figure 5.4

Generally the same IVS take the same commuting route every day during rush-hour. Because these are ideal conditions for an attacker to link pseudonyms applied

[1] www.OpenStreetMap.org

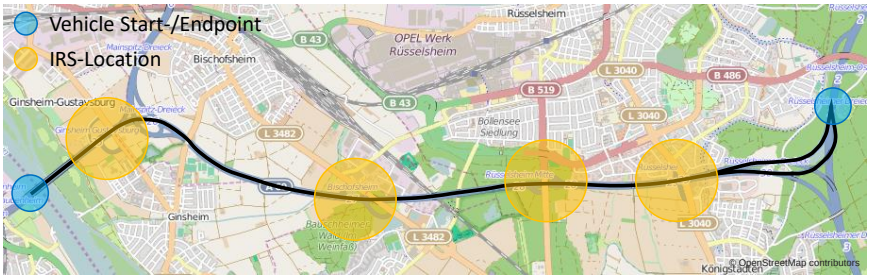


Figure 5.4.: Simulation scenario

at different days to the same IVS, we evaluated the protocol under these conditions. Other scenarios without recurring ITS Vehicle Stations or IVSs on different routes make it more difficult for an attacker to link the identities. Furthermore, we assume an IVS changes its pseudonyms at the beginning of each day.

According to the traffic density categorization in [For05] we applied a high density of IVSs in one and a low density in the other direction. This is common for rush-hour when most people aiming towards the city and only a few drive outbound. Three classes of IVSs are considered in the simulation: The fast ones have a maximum speed of 130 km/h, the regular ones a maximum speed of 110 km/h, and the slow ones of 80 km/h. The different classes are equally distributed. The IVSs only drive the maximum speed if the traffic conditions allow it. They also overtake only if there is space to do so. Furthermore, only ten percent of the IVSs are equipped with an application software that utilizes the proposed anonymous key agreement protocol. This is a typical market share for big car manufactures [Blo14]. More equipped IVSs would make it even more difficult for an attacker to link the identities. We assume the pseudonyms of all IVSs have the same validity duration, otherwise an attacker can exploit this information to identify an IVS.

The envisaged simulation duration is 60 minutes. Since it takes some time until the simulation is adjusted, we cut 10 minutes both at the beginning and at the end of the simulation. Due to the long simulation duration for one run, we decided to run the simulation without considering the different protocol parameters detailed in Section 5.4.2. Instead, we log which IVSs establish a session key during the simulation and map all parameters afterwards in the IVSs. We run 50 simulations to create a sufficient pool of simulation results to choose from. To evaluate more than 50 days, we randomly select as much simulation results as necessary from the pool of all 50 simulation runs and map the protocol parameters afterwards into the simulation.

The elaborated results show that every day each IVS executes the protocol at least

Table 5.2.: Range of the considered parameters

Parameter	Range	Default
Ring size	2 - 15	10
One time vehicles	10% - 90%	30%
Starting time deviation	1 min - 10 min	5 min
Ring building strategy	all outlined	SameDirectionLastDays
Parallel pseudonyms	1 - 15	10
Attack duration	1 day - 360 days	30 days

once in the communication range of each IRS. Furthermore, we assume an attacker on the plain version of the protocol. Thus, an attacker needs only one IRS under her control in order to record the pseudonyms applied in the rings of all passing IVSs. Therefore, an attacker also cannot get more information about the applied pseudonyms by having control over some of the IVSs. In Section 5.4.3 we describe the behavior of the attacker in more detail.

5.4.2. Considered Parameters

The anonymity of the IVS is influenced by various parameters when they apply the proposed anonymous authentication protocol. We considered the following parameters in the subsequent simulation runs. A summary of all parameters and their range of values is given in Table 5.2.

Ring size

The ring size denotes the number of pseudonyms present in the ring signature. We varied the ring size between 2 and 15 to assess the impact. Unless explicitly mentioned, we apply 10 pseudonyms.

Fraction of one time vehicles

These IVSs apply a set of pseudonyms in their ring that is completely unknown to the other IVSs. They shall reflect that most IVSs drive the same route each day, but there are always IVSs that normally do not take this route in rush-hour, e.g., trucks. We varied this value between 10% and 90% to determine the effect of this parameter. Unless explicitly mentioned, we consider 30% of such one time vehicles.

Standard deviation of the starting times

People are driving to work every day at approximately the same time. Therefore, the starting times of the IVSs are assumed to be normally distributed. The standard deviation has an influence on the potential communication partners. Therefore, we alter it between 1 and 10 minutes. Unless explicitly stated we use a standard deviation of 5 minutes.

Ring building strategy

When ring signatures are in place, an IVS is one of n possible signers. It is important to apply a good ring building strategy, because a poor strategy can lead to revealing of most or even all of the non-signers, so the anonymity of the signer decreases. In the following we suggest some appropriate strategies to build a ring. We evaluate these strategies later on in Section 5.4.4.

All: The IVSs collect and save all pseudonyms they receive from other IVSs. When the IVSs need to build a new ring, they randomly select the required number of pseudonyms from their pools.

SameDirection: IVSs using this strategy collect and save all pseudonyms they receive from other IVSs driving in the same direction. The basic idea behind is that IVSs in rush-hour drive every day at approximately the same time in the same direction. Therefore, an attacker cannot delete the pseudonyms of the IVSs driving each day in the opposite direction from the ring. The same ring building strategy as for "All" is applied.

SameDirectionLastDays: This strategy is similar to "SameDirection". The only difference is that the IVSs discard pseudonyms they met more than X days ago. The reason for this is that each day the IVSs collect pseudonyms of one time vehicles they never met before and unlikely meet again. If an IVS applies such pseudonyms in its ring, an attacker is able to identify and remove them to get the identity of the victim IVS. This can be done because they are less used than other pseudonyms. When limiting the number of pseudonyms by the number of previous days, the influence of these IVSs decreases. Unless explicitly mentioned, we apply this ring building strategy. To get the optimal number of previous days, we ran simulations with day values ranging from 1 to 10 for the various pseudonym pool sizes. Based on the outcome of these runs, we selected the most appropriate number of days.

SameDirectionLastDaysDifferentSizes: This strategy works similar to "SameDirectionLastDays", but each IVS applies an own ring size. This strategy is evaluated later on to assess the influence of different ring sizes on the anonymity of the IVSs.

Number of own pseudonyms

The number of own pseudonyms an IVS possesses at the same time is an important parameter for the privacy. Each time a ring is being build, the IVS randomly selects one. We varied the number of parallel pseudonyms between 1 and 15 to assess the impact. Unless explicitly stated, we exploit 10 simultaneous pseudonyms.

Duration of the attack

The duration of the attack denotes the number of days the attacker listens to the exchanged messages. We vary the duration between 1 and 360 days to evaluate the effect of this parameter. Unless explicitly mentioned, we consider an attack duration of thirty days.

5.4.3. Attacker Behavior

The considered attacker in the simulation is a passive one aiming at the plain version of the protocol. Furthermore, the attacker has knowledge of all simulation parameters. This attacker type is sufficient, because even the most powerful attacker aiming at the encrypted or plain version cannot gain more information. The attacker tries to identify the application-specific pseudonym of an IVS from the ones used in the ring signature. The behavior of the attacker can be divided into three stages.

In the first stage the attacker records all exchanged messages in order to analyze them later on.

After recording, the attacker counts how often each pseudonym has been applied. Then, she selects the relevant pseudonyms. These are the ones which are applied at least $DaysObserving/PseudonymPoolSize$ times, where *DaysObserving* denotes the number of days the attacker recorded the messages and *PseudonymPoolSize* the number of valid own pseudonyms each IVS possesses at one point in time, respectively. Thus, only pseudonyms applied regularly are considered. The filtered ones might be introduced to the communication by IVSs driving only once the observed street. The pseudocode for the filtering algorithm is given in Figure 5.5.

The third and last stage starts with the identification of unambiguous pseudonyms and is shown as pseudocode in Figure 5.6. A pseudonym is unambiguous, if it is the only relevant pseudonym of a ring. Therefore, this pseudonym must be the identity of the IVS.

```
1: // Count pseudonym usage
2: Create hashmap hm for pseudonym usage
3: for each collected message m do
4:     for each used pseudonym p in m do
5:         if hm.contains(p) then
6:             hm.put(p, hm.get(p) + 1)
7:         else
8:             hm.put(p, 1)
9:         end if
10:    end for
11: end for
12: // Select relevant pseudonyms
13: Create list relevantPseudonyms
14: for each pseudonym p in hm do
15:    if p > DaysObserving/PseudonymPoolSize then
16:        relevantPseudonyms.add(p)
17:    end if
18: end for
```

Figure 5.5.: Pseudonym filtering algorithm applied by the passive attacker

Afterwards, these unambiguous pseudonyms are deleted from all rings of the other IVSs on this day. By 'delete' we mean that it is now clear that this pseudonym does not belong to the IVS and we therefore do no longer need to consider it in the respective rings.

If each IVS applies an own ring size, unambiguous pseudonyms are also deleted from the IVSs applying a different ring size in other days. By 'own ring size' we mean that not all IVSs apply the same number of pseudonyms to construct their ring signature. We can delete these pseudonyms, because we know the ring size of the IVS owning the pseudonym is different.

If any pseudonyms were deleted, the attacker tries to identify new unambiguous pseudonyms, otherwise the attacker is finished.

Now the attacker has reduced the ring size of the IVSs by excluding pseudonyms, which cannot be the identities of the IVSs. We evaluate in the sequel by how much the attacker is able to reduce the ring size and therefore the k-anonymity in presence of the different parameters.

```
1: do
2:     reduced = false
3:     // Identify unambiguous pseudonyms
4:     Create list unambiguousPseudonyms
5:     for each collected message m do
6:         for each pseudonym p in m do
7:             if m.relevantPseudonyms == 1 then
8:                 unambiguousPseudonyms.add(p)
9:             end if
10:        end for
11:    end for
12:    // Reduce ring sizes
13:    for each collected message m do
14:        for each pseudonym p in m do
15:            if p ∈ unambiguousPseudonyms ∧ m.day ==
   unambiguousPseudonyms.get(p).day then
16:                m.delete(p)
17:                reduced = true
18:            end if
19:        end for
20:    end for
21:    // Handle different ring sizes
22:    if differentRingSizes then
23:        for each collected message m do
24:            for each pseudonym p in m do
25:                if p ∈ unambiguousPseudonyms ∧ p.ringSize ! = m.ringSize then
26:                    m.delete(p)
27:                    reduced = true
28:                end if
29:            end for
30:        end for
31:    end if
32: while reduced == true
```

Figure 5.6.: Ring size reduction algorithm applied by the passive attacker

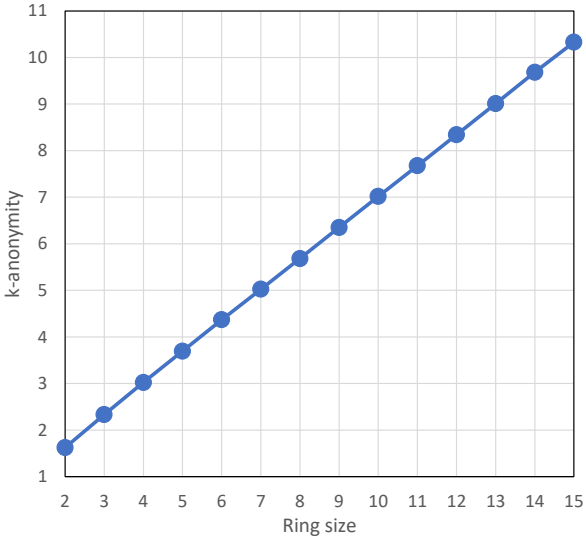


Figure 5.7.: Impact of the ring size to the k-anonymity

5.4.4. Influence of Considered Parameters

In this section we discuss the evaluation results of the previous introduced parameters and give recommendations on suitable values.

Ring size

The k-anonymity of the IVSs, calculated according to [Swe02], increases linearly with the ring size from 1.6 when a ring size of 2 is applied over 7.0 when a ring size of 10 is used, up to 10.3 when a ring size of 15 is being applied. The values for all ring sizes from 2 to 15 are illustrated in Figure 5.7. A larger ring size increases the number of potential signers. Subsequently, it gets more difficult for an attacker to identify the creator of the signature. Therefore, we recommend using a ring size as large as possible.

Fraction of one time vehicles

The k-anonymity of the IVSs decreases from 8.6, when only 10 % of the IVSs are one time vehicles over 5.5, when 50 % of the IVSs are one time vehicles, down to 2.7, when 90 % of the IVSs in the simulation appear only once. The reason for the

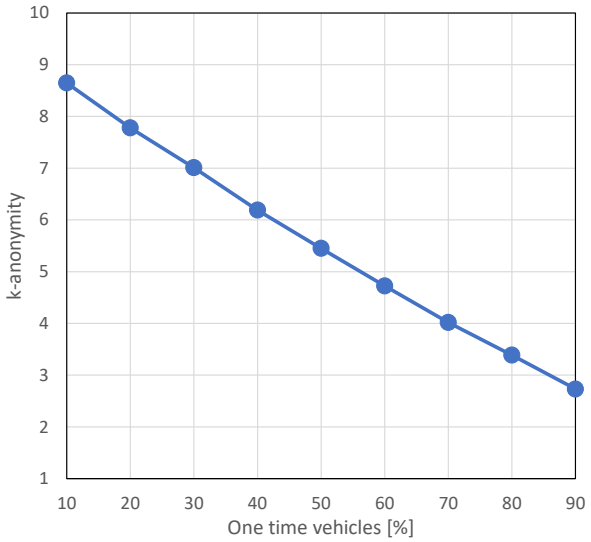


Figure 5.8.: Impact of the number of one time vehicles to the k-anonymity

decrease is that there are more new IVSs and therefore also more new pseudonyms in the simulations, which are considered by the IVSs during ring building. These values are also illustrated by Figure 5.8.

Standard deviation of the starting times

An increase or decrease of the standard deviation of the starting times had no notable influence on the anonymity of the IVSs. However, the starting times of an IVS for the default standard deviation of 5 minutes and 1000 days is given in Figure 5.9.

Ring building strategy

The influence of the ring building strategy to the k-anonymity is shown in Figure 5.10. The x-axis displays the month in which the attacker analyzes the messages since the start of pseudonyms usage. Month 1 is therefore the analysis of the first month, month 2 of the second month, and so on.

The k-anonymity value decreases over time for the strategies "All" and "SameDirection". The strategies "SameDirectionLastDays" and "SameDirectionLastDaysDifferentSizes" maintain the k-anonymity of the IVSs after the second month.

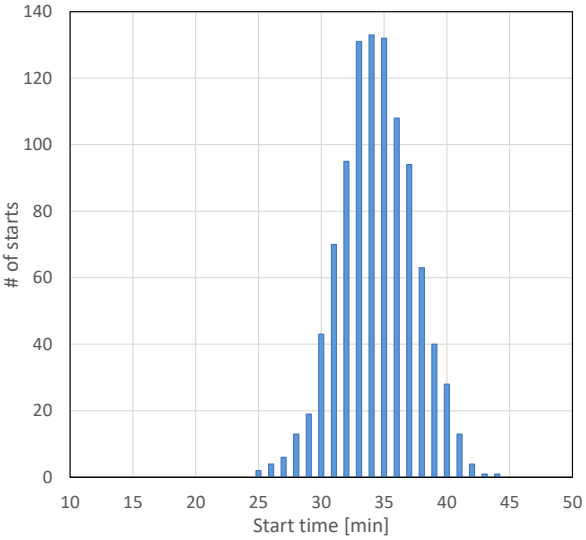


Figure 5.9.: Example starting times for a standard deviation of 5 minutes

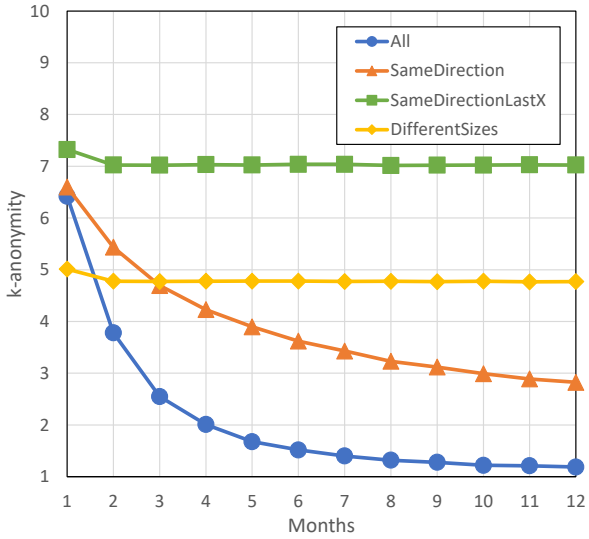


Figure 5.10.: Impact of ring building strategies

For "All" the k-anonymity decreases from 6.4 in the first month down to 1.2 in the twelfth month. This is because of the potential high number of pseudonyms from IVSs driving in opposite direction applied in the ring signatures of the IVSs. Therefore, an attacker is able to exclude them from the list of possible signers.

The average k-anonymity value when using the "SameDirection" strategy is 6.6 in the first month and steadily decreases over time down to 2.8 after twelve months. This strategy decreases slower than "All" because the IVSs applying it do not include IVSs from the opposite direction in their ring signatures. However, the attacker is still able to filter out the applied pseudonyms from one time IVSs.

When applying the "SameDirectionLastDays" strategy, the IVSs have a constant k-anonymity of 7.0 from the second month on. This is because the IVSs apply only pseudonyms obtained in the last few days to create their ring signature. Therefore, it is even more difficult for an attacker to filter out irrelevant pseudonyms. Furthermore, the number of irrelevant pseudonyms does not increase over time.

For the strategy "SameDirectionLastDaysDifferentSizes" the k-anonymity value drops from 7.0 to 4.8 compared to the case when the same sizes are used after the first month. When IVSs apply different ring sizes, it is possible for an attacker to further reduce the number of possible signers by excluding the pseudonyms applied in signatures with a different ring size.

These results show that all IVSs should apply the same ring size to keep the k-anonymity at a high level. If only pseudonyms received in the last few days are considered, the k-anonymity of the IVSs does not decrease over time.

Number of own pseudonyms and duration of the attack:

Figure 5.11 illustrates the k-anonymity of the IVSs for different numbers of own pseudonyms as a function of the number of days an attacker records the exchanged messages. It shows that the k-anonymity value decreases with the number of days an attacker listens to the exchanged messages. With each additional day the attacker gets more information to identify new irrelevant pseudonyms and remove them from the ring signatures. In addition, the k-anonymity value increases with the number of own pseudonyms. When the IVSs possess less pseudonyms valid at the same time there are less circulating pseudonyms and it gets easier for an attacker to identify irrelevant pseudonyms. Depending on the assumed attack duration, either more pseudonyms have to be used or the pseudonyms have to be renewed more often in order to maintain a certain level of anonymity.

If the operator of an application aims, for example, at a k-anonymity of at least 5 for its subscribers, the IVSs could apply a new set of 5 pseudonyms every 30 days, a set of 10 pseudonyms every 60 days or a set of 15 pseudonyms every 90 days. This sums up to 60 pseudonyms per year. If we compare this value to the

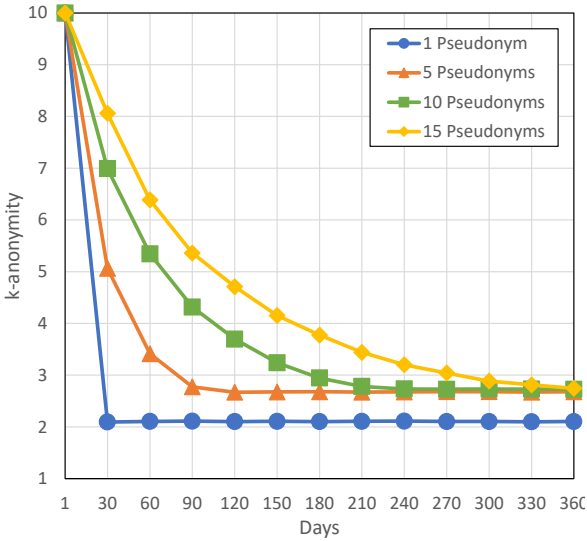


Figure 5.11.: Impact of the own pseudonym pool size and attack duration

safety pseudonyms, where the pseudonym should be changed at least with every trip, we can easily see that the proposed protocol reduces the number of necessary pseudonyms considerably. The amount and usage duration of pseudonyms also depends on the online connectivity and available storage of the IVSs.

We showed in this section that the k-anonymity of the IVSs can be enhanced by the protocol. At the same time, it reduces the number of pseudonyms each IVS needs to store considerably. Furthermore, we showed that parameters like the ring size or number of parallel pseudonyms have a huge influence on the resulting k-anonymity of the IVSs.

5.5. Real-World Evaluation

The outlined protocol was also evaluated on vehicles equipped with real-world hardware. We created a prototype implementation to evaluate the size of the ring signature, size of the transmitted messages, and the time necessary to execute the whole protocol, each single step, and the different phases. Furthermore, we evaluated how the prototype implementation is affected by faulty messages, the execution with multiple IVSs at the same time and the exchange of real payload.

5.5.1. Implementation

We outline the prototype implementation of the key-agreement protocol in the sequel. The implementation was done as part of [Bar15a]. We applied Java as programming language to implement the protocol since we only had access to a GeoNetworking implementation in Java. For cryptographic operations we applied the Bouncy Castle[2] library. As ring signature scheme we implemented the version proposed in [LLZ+07]. Message size is an important factor in VANETs because the channel capacity is very limited. In order to minimize the channel load, we applied EC point compression [VMA00]. Furthermore, we used the Java *Deflater* to compress the pseudonyms applied in the ring signature with zlib[3]. The required certificates were issued by the Car2Car Pilot PKI from the C2C-CC, which are compliant with the corresponding ETSI standard [ETS13]. The pseudonyms applied for signing the GeoNetworking package contain one key for signing and one for encrypting data each. Furthermore, the AIDs for CAMs and DENMs were included. This resulted in a total pseudonym size of 170 bytes. For the application specific pseudonyms exploited by the ring signature we applied 42 as AID and a key for signing. This resulted in a total size of 132 bytes for the application specific pseudonyms.

Because of the frequent topology changes in VANETs, IVSs often leave the communication range of temporal communication partners. Therefore, we addressed such communication losses as follows. If a message was not received by the other IVS, both partners run in a timeout and the execution of the protocol is stopped. We did not implement retries, because this could result in an unnecessary channel congestion due to the fact that the other IVS is most likely no longer in the communication range. If an error is detected in a message, the execution of the protocol is also stopped. If the IVSs detect afterwards that they are still in communication range, they execute the protocol from scratch. If the execution of the protocol fails more than n times with one IVS, no further attempts to execute the protocol are made. Only messages relevant to execute the protocol are processed. Other messages like CAMs are dropped. The protocol is only executed once with each IVS in a configurable time interval. The corresponding state machine of the protocol implementation is illustrated in Figure 5.12.

To support the parallel execution of the protocol with multiple IVSs at the same time, the implementation has a dedicated task to send the periodical SAM. A second task processes all incoming messages. In a first step it drops the messages not relevant for the protocol execution. Then it checks if the protocol is already executed with the IVS. In case the protocol is not yet executed, it creates a new state machine for this IVS and passes the message to it. If the protocol is already executed with the

[2] www.bouncycastle.org
[3] www.zlib.net

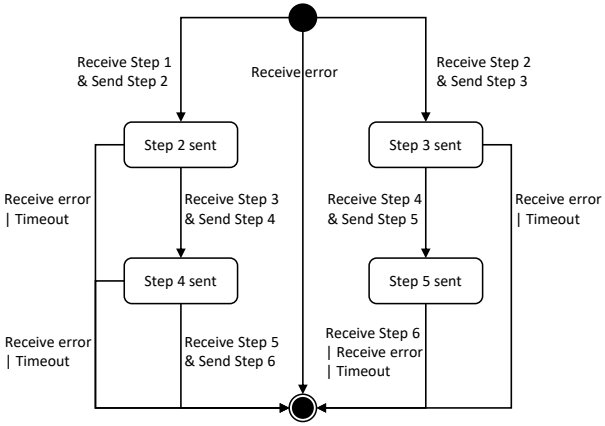


Figure 5.12.: State machine of the protocol implementation

IVS, the message is directly passed to the corresponding state machine.

The SAM rate, timeout, and number of failed protocol executions until the execution is stopped can be configured as a parameter at startup. The implementation also supports different ring sizes and arbitrary payload. Furthermore, we utilized the first byte of the payload to indicate the version and protocol step number.

5.5.2. Measurement Setup

The measurement setup for the real-world evaluation consists of multiple development vehicles. Each vehicle is equipped with one Application Unit (AU) and one Communication and Control Unit (CCU). The AU runs the application software. As hardware for the AU we utilized a NEXCOM VTC6200 with an Intel Atom D510 Dual Core CPU with 1.6 GHz and 2 GB of memory. As operating system Ubuntu Linux 14.04 was applied. The CCU is responsible to send messages over DSRC to other vehicles. As CCU we utilized a NEC LinkBird-MX, which has a MIPSel architecture and Debian etch as operating system. The messages generated by the AU are forwarded via a User Datagram Protocol/Internet Protocol (UDP/IP) packet to the CCU, which sends them on reception over DSRC to all IVSs in communication range. Once the message is received by the CCU of another IVS, it is forwarded via a UDP/IP packet to the AU. There, the message is processed and a response generated. This setup is illustrated by Figure 5.13. In order to keep the evaluation procedure simple, only two vehicles were used to measure the execution times of the protocol if not otherwise mentioned.

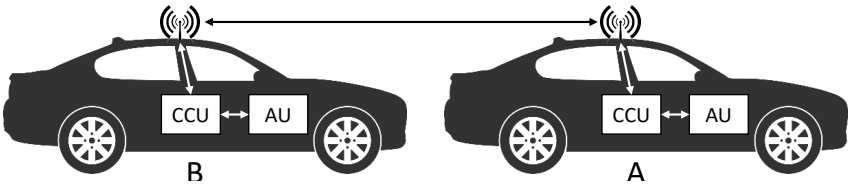


Figure 5.13.: Protocol evaluation setup

5.5.3. Signature Size

The signature size needs careful consideration. For the driver it is important to aim at a ring signature as big as possible and, therefore, to be indistinguishable from as many other signers as possible. However, each additional signer of a ring signature increases the size of the ring signature by one pseudonym and one elliptic curve point. Since we want the maximum possible anonymity for the IVSs, we try to make the ring as large as possible, but we also have to avoid any message fragmentation at the Medium Access Control layer if possible [ETS09a]. Therefore, we analyze the size of the signature depending on the number of ring members first and determine the maximum feasible message size later in Section 5.5.4.

As described in Section 5.5.1, we exploited the EC ring signature scheme proposed in [LLZ+07] with curve P-256 and pseudonyms from the Pilot PKI of the C2C-CC. Furthermore, we applied point compression and an AID with SSPs to bind them to a specific application. The resulting size of each pseudonym is 132 bytes. The resulting signature size as a function of the ring sizes is shown in Figure 5.14. The size of the signature increases linearly with every ring member by the size of one pseudonym *Cert* (132 bytes) and one EC-Point x (33 bytes), which are in total 165 bytes.

5.5.4. Message Size

Because some messages of the protocol increase with the ring size, we have to determine the largest message in the protocol and measure its size to determine the maximum possible ring size while avoiding message fragmentation. The largest message in the proposed protocol results from Step 1. It contains all ECIES parameters and a ring signature. However, the pseudonyms applied in the ring signature might be compressed in order to reduce the overall size of the message.

The size of the complete GeoNetworking packet for the first step of the protocol as a function of the ring sizes for both versions of the protocol with and without compression is illustrated by Figure 5.15. We observed that the Linux software driver

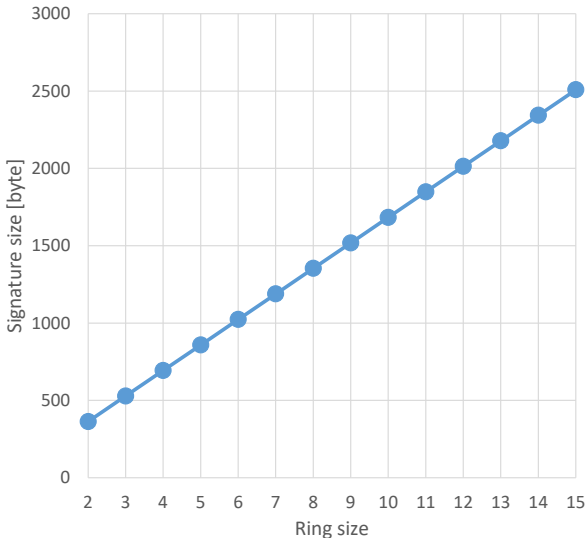


Figure 5.14.: Signature size as a function of the ring size

of the DSRC card within the CCU does only support a maximum message size of 2364 bytes when the largest possible Maximum Transmission Unit (MTU) is in operation. Therefore, the grey area of the figure indicates the maximum message size of the applied hardware. These measurements are done on a computer with the same implementation of the protocol as used later in real vehicles in order to measure larger messages.

The message size increases linearly with the ring size. In addition to the signature size evaluated in Section 5.5.3 it consists of the ECIES parameters V (65 bytes), C (20 bytes), and T (16 bytes), the GeoNetworking information (322 bytes) as well as the protocol step information (1 byte). It furthermore shows that compression reduces the message size considerably. However, there is no huge difference between the message of the plain and encrypted version of the protocol.

The graph indicates that for the maximum packet size of 2364 bytes as for the utilized hardware a ring size of 11 is feasible if no compression is applied. When compression is in operation and fragmentation at the Medium Access Control layer shall be avoided a ring size of 13 is possible for the plain and encrypted version of the protocol. Although 13 does not denote a large set, we showed in Section 5.4 that even a ring size of 10 is sufficient for the protocol and use case presented in order to preserve the privacy of the IVSs while reducing the number of necessary

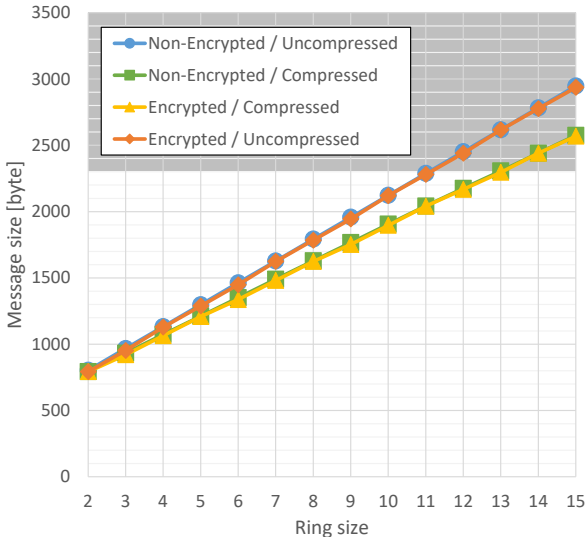


Figure 5.15.: Message size of the complete GeoNetworking packet for the second step both versions of the protocol as a function of the ring size with and without compression

pseudonyms. For the feasible ring sizes between 11 without compression and 13 with enabled compression the payload of the GeoNetworking package is around 81 % of the overall message size.

Messages of this size are much larger than safety messages. However, they are sent at a much lower frequency. Safety messages are sent up to 10 times a second, whereas the messages in the proposed protocol for the envisaged use case will be sent only a few times per hour. The messages are also sent on a different channel as the safety messages and do not have critical time constraints like safety messages. The envisaged channel is expected to be also applied for Internet browsing, video streaming or software updates.

The influence of compression for the different ring sizes is depicted in Table 5.3. It shows that the compression factor increases with the overall message size. For a ring size of 13 the overall message size is reduced by over 12 % when the encrypted version of the protocol is applied.

The message size for the plain and encrypted version of the protocol are similar. They only differ by the padding bytes of AES. We have a closer look to this in the sequel. Figure 5.16 shows the difference of the largest message of the encrypted

Table 5.3.: Compressed and uncompressed message size for both protocol versions in bytes and the gained compression

Ring Size	Uncompressed		Compressed		Compression Gain	
	plain	enc	plain	enc	plain	enc
2	802	792	793	792	1.12 %	0.00 %
3	967	952	932	920	3.62 %	3.36 %
4	1132	1128	1073	1064	5.21 %	5.67 %
5	1297	1288	1211	1208	6.63 %	6.21 %
6	1462	1448	1350	1336	7.66 %	7.73 %
7	1627	1624	1490	1480	8.42 %	8.87 %
8	1792	1784	1628	1624	9.15 %	8.97 %
9	1957	1944	1766	1752	9.76 %	9.88 %
10	2122	2120	1905	1896	10.23 %	10.57 %
11	2287	2280	2042	2040	10.71 %	10.53 %
12	2452	2440	2175	2168	11.30 %	11.15 %
13	2617	2616	2306	2296	11.88 %	12.23 %
14	2782	2776	2440	2440	12.29 %	12.10 %
15	2947	2936	2575	2568	12.62 %	12.53 %

and plain version in bytes. The negative values indicate that the encrypted version of the protocol has a smaller message size. When the plain version is applied only the signature is encrypted. It has a length of 33 bytes and is therefore encrypted in three AES blocks of 16 bytes, where the last block is padded with 15 bytes to a multiple of the block size. In the encrypted version of the protocol more data is encrypted, whereby less padding bytes are applied and therefore the overall message size is reduced. Therefore, the message size of both versions differ by a maximum of 15 bytes. For the maximum possible ring size of 13, the message size of the encrypted version is 10 bytes smaller than for the plain version.

The lengths of the single parts of the GeoNetworking message as function of the ring size (rs) and payload length (pl) are depicted in Table 5.4 for both protocol versions and all steps.

Because the encrypted version of the protocol has, as shown in the simulation, a better protection from attackers and compression enables the usage of a larger ring size, we apply these features in the following if not otherwise stated.

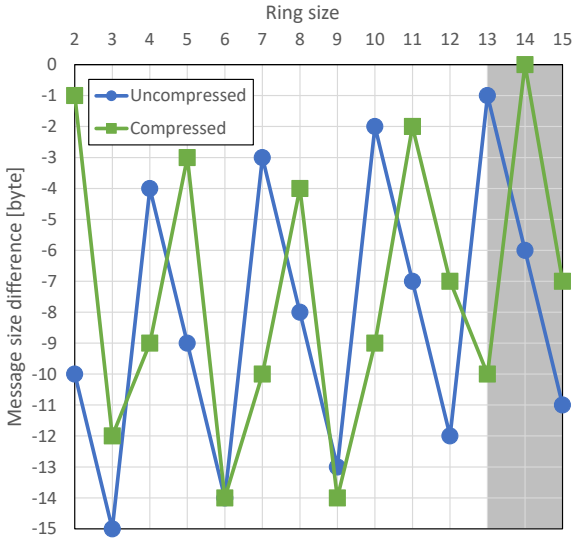


Figure 5.16.: Message size as a function of ring size: Size difference of encrypted and plain message

Table 5.4.: Lengths of the single GeoMessage parts for the different protocol steps in bytes for both protocol versions without compression

Part	Step 1	Step 2	Step 3 / 4	Step 5
BH	4	4	4	4
SP Info	2	2	2	2
HF	183	183	183	183
PF Info	5	5	3 for $pl <= 64$ 5 for $pl > 64$	3
CH	8	8	8	8
EH	48	48	48	48
TP	4	4	4	4
Payload plain	$1+101+192*rs+48$	$1+192*rs+48$	$1+16*\lceil\frac{pl}{16}\rceil$	1
Payload encrypted	$1+101+16*\lceil\frac{192*rs+33}{16}\rceil$	$1+16*\lceil\frac{192*rs+33}{16}\rceil$	$1+16*\lceil\frac{pl}{16}\rceil$	1
TF	68	68	68	68

5.5.5. Execution Time

We measured the execution time of the complete protocol, of constituent steps, and of the different phases to assess the feasibility of the protocol in terms of the execution time. For each measurement, we executed the protocol 550 times. If not otherwise mentioned, we used a payload length of 1.000 bytes, the encrypted version of the protocol, compression, a SAM rate of 3 seconds, a timeout of 3 seconds, and a protocol execution repetition of 5 seconds. We exploited random bytes as the payload in Steps 4 and 5 if not otherwise stated. Due to code optimizations of the Java Virtual Machine (VM) at run time, we observed a steady reduction in the total execution time of the protocol for the first executions of each measurement. Therefore, we dropped the first 50 measurements of each run.

Complete Protocol

The measurement for the execution time of the complete protocol starts when IVS A sends the SAM by its AU and ends when the same IVS receives the acknowledgment of Step 5 by the AU. We measured this time for both protocol versions with compressed and uncompressed pseudonyms and different ring sizes. We also varied the amount of payload between 100 and 1.500 bytes and the speed of the IVSs between 0 and 130 km/h. The results of these measurements are discussed in the sequel.

Because of the maximum supported message size of 2364 bytes for the exploited DSRC card we were able to measure the execution times for ring sizes with 13 or less signers only. The resulting execution time for the protocol increases linearly depending on the ring size value. For example, a ring size of 2 requires for the whole protocol execution 188 ms time, while it takes around 412 ms to complete the protocol for a ring size value of 13.

The increase of the protocol duration for a larger ring size is due to the fact that for a larger ring size more cryptographic operations have to be executed in order to create or validate a ring signature. In addition, more transmission time is necessary to transmit these larger messages. The operations to create and validate a ring signature are executed twice, i.e., once for each participating IVS. If the time necessary to create and validate the ring signatures is removed from the overall execution time, then the execution time increases only a few milliseconds for each additional ring member. This slight increase is due to the additionally transferred data. The overall execution time, the time without the ring operations as well as the time necessary to create or validate the ring signatures are shown in Figure 5.17.

In order to rate the complete execution time of 412 ms when a ring size of 13 is applied, it is necessary to know how long IVSs stay in communication range. In general, IVSs in a VANET have a communication range of several hundred meters [dB14]. If we assume two IVSs having a communication range of only 100 m

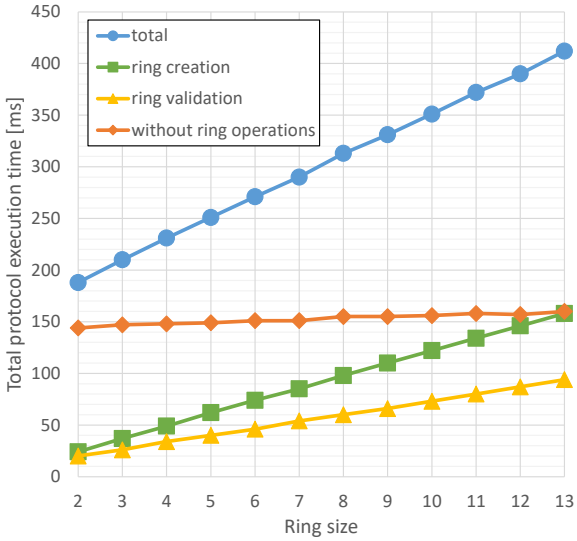


Figure 5.17.: Total execution time of the protocol, execution time without the ring operations and the duration of the ring operations for different ring sizes

and a relative velocity of 200 km/h, i.e., the IVSs are driving in opposite directions each with a speed of 100 km/h, then they stay in their communication range for more than 3 seconds. Therefore, there is plenty of time to execute the protocol, even under such harsh conditions.

When the payload transmitted in Steps 4 and 5 is increased, the protocol needs only a few additional milliseconds to be executed because of the larger message sizes. Different speed of the IVSs has no measurable influence on the execution time of the protocol. Furthermore, we did not observe any difference resulting from executing the encrypted or the plain version with or without compression.

Protocol Steps

We measured how long it takes to execute each step of the protocol. In order to do so we started the measurement when the message of the previous step is received by the application software running on the AU until the new generated message is sent from the Java application to the CCU. Therefore, the duration of each step includes the verification of the received GeoNetworking package and creation of the sent GeoNetworking package. We calculated the time necessary to transmit the message between the IVSs from the information how long each single step takes and from the complete execution duration.

The results illustrate the influence of the ring signature operations on the overall execution time as visible in Figure 5.18. Step 1 to 3 consume 80 % of the overall protocol execution time. Step 1 is dedicated to the creation of a ring signature. In Step 2 the resulting ring signature is validated and a new ring signature is created. In Step 3 the verification of the ring signature created in Step 3 is performed. All other steps, the transmission and other operations take 20 % of the execution time. A comparison of the execution duration of the protocol steps for different ring sizes is given in Appendix A.1.

Figure 5.19 shows the time necessary to execute the individual parts of the single protocol steps. We differentiate between *Sending*, *Receiving*, *AES*, *ECIES*, *Ring Creation*, *Ring Verification*, and *Other*. Each step includes *Receiving* which is the time to parse the GeoNetworking message and verify its signature. Furthermore, all steps except the verification of Step 5 consume time for *Sending* where the new GeoNetworking package is built and a corresponding signature is generated. *AES* and *ECIES* denote the respective cryptographic operations. *Ring Creation* is the time necessary to create a new ring signature while *Ring Verification* is the time necessary to verify a ring signature. The remaining operations are grouped in *Other*.

For all steps *Sending* always takes between 5 and 6 ms, while *Receiving* takes between 10 and 12 ms. The *AES* operations for encryption and decryption take no measurable time. The main time consumed in the Steps 1 to 3 is caused by the ring

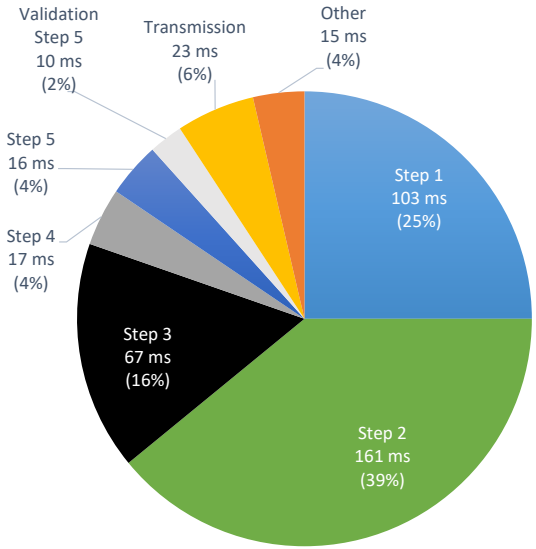
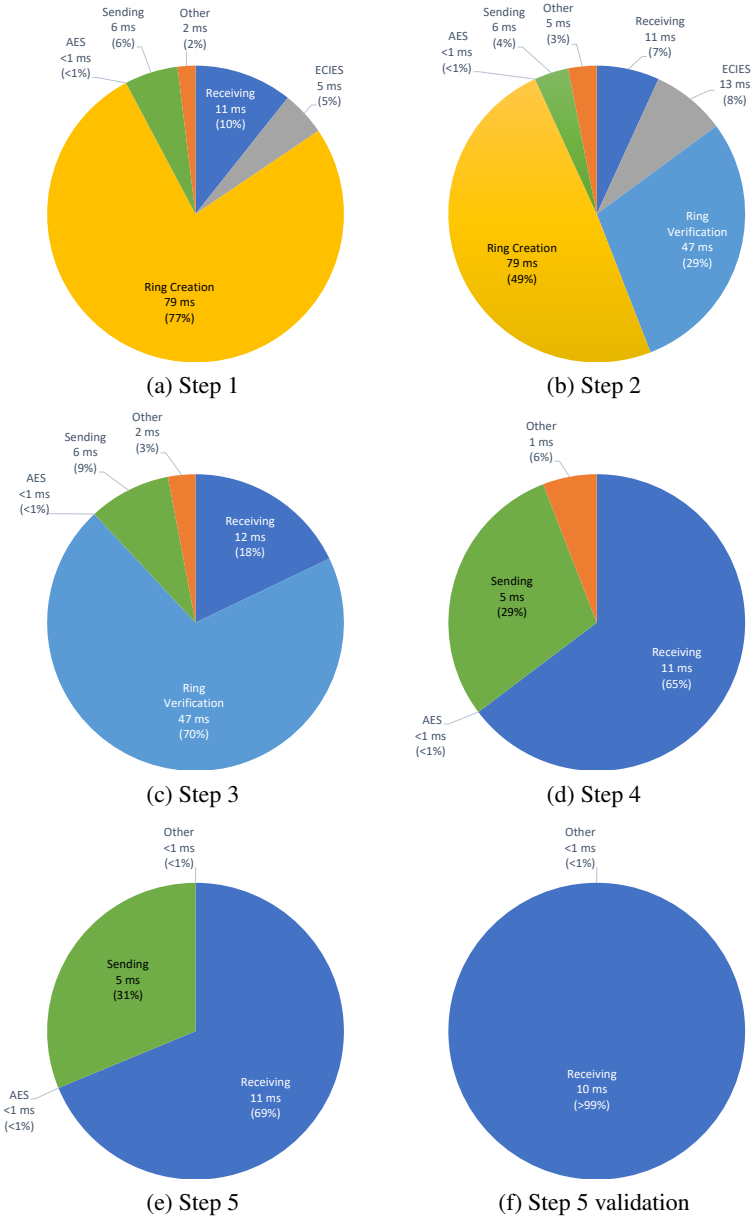


Figure 5.18.: Duration of the protocol steps, transmission time, and other operations

signature operations *Ring Creation* and *Ring Verification* with at least 70 % in each step. The execution times of the single parts of the protocol steps for ring sizes ranging from 2 to 12 are given in Appendix A.1.

Protocol Phases

The duration of the protocol can also be split into different protocol phases, namely *Security*, *Transmission*, *Receiving / Sending*, and *Other* minor operations. The security phase of the protocol cause nearly two third of the overall execution time as visible from Figure 5.20. The GeoNetworking operations *Receiving / Sending* take in total 94 ms, which is nearly a quarter of the whole execution time. These operations contain the security operations to create and verify the GeoNetworking packet. We included the security in there, because we assume GeoNetworking, including its security, as given. Therefore, it is independent of the security the key agreement protocol introduces. However, in our evaluation setup the security is responsible for 66 ms of the overall 94 ms of the execution time for GeoNetworking. This time will be reduced considerably in production vehicles. There it will be implemented in hardware to meet the timing requirements [SAE16b]. The remaining execution time is shared between *Transmission* and *Other*.



(a) Step 1 (b) Step 2

(c) Step 3 (d) Step 4

(e) Step 5 (f) Step 5 validation

Figure 5.19.: Necessary time to execute the individual parts of the protocol steps

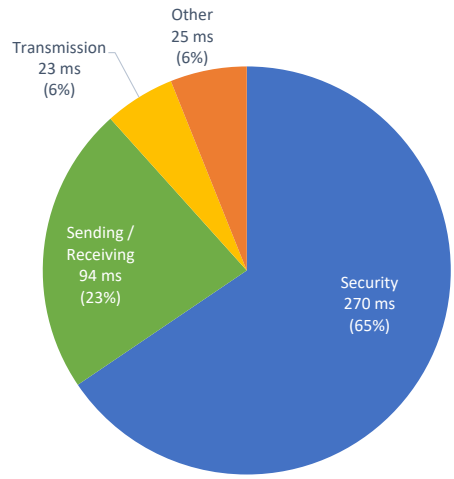


Figure 5.20.: Duration of *Security*, *Receiving / Sending*, *Transmission*, and *Other* times in percent of the total execution time

The security might be further split into *Ring Signature* operations, *ECIES*, and *AES* as illustrated by Figure 5.21. This shows that the creation and validation of the ring signatures consume 93 % of all time necessary for security operations. *ECIES* takes only 7 % of the security time while the time for *AES* is negligible. A considerable time consumed for security purposes could be reduced either by creating a more efficient software version or by mapping the algorithms to a dedicated hardware implementation.

5.5.6. Faulty Messages

We created a piece of software, which allows us to inject bit errors in certain parts of the message or to change the order of the received and sent messages. Thus, we are now in the position to test the robustness of the implemented protocol. We injected random bit errors in the ECIES parameters, ring signature, payload, or in the whole secured GeoNetworking packet. The outlined implementation of the protocol successfully detected all introduced errors during the message processing. Furthermore, it always recovered successfully from the resulting error states by executing the protocol from scratch.

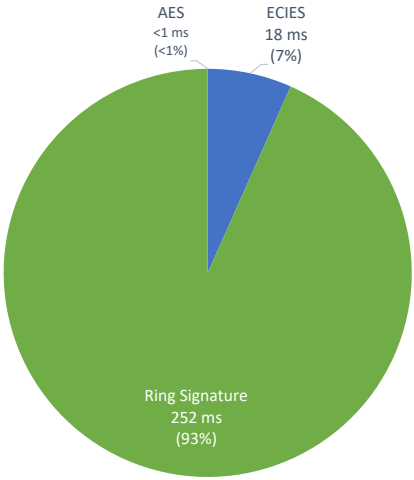


Figure 5.21.: Duration of different security operations

5.5.7. Multiple Communication Partners

All previous presented measurements are based on the execution of the protocol between two IVSs. In a real-world scenario an IVS might not only meet one ITS Vehicle Station at a time to execute the protocol. It rather may in communication range of multiple IVSs supporting the protocol. Therefore, it should be possible to execute the protocol with multiple IVSs at the same time. We tested this by running our implementation on three hardware setups at the same time. Two IVSs run the software without sending SAMs. Afterwards the program is started on the third hardware setup with activated SAMs. As soon as the other two IVSs received the first SAM, they started sending protocol Step 1 to the initiating hardware setup, which then responded to both IVSs separately with Step 2. This was also done with the remaining protocol steps until all steps are transmitted. Therefore, these results show that it is possible to execute the protocol with multiple IVSs at the same time. Hence, an IVS is able to confidential exchange data with different IVSs at the same time by applying the outlined protocol.

5.5.8. Real Payload

In the previous tests random data was applied as payload in Step 3 and 4. We additionally exploited an application which utilized the built-in camera of the IVS in

order to detect and store the passing traffic signs locally [Bre14] as an example application to exchange real data between IVSs. The transmitted data included the detection time, type, and position including the altitude and heading of the traffic sign. We exchanged the traffic signs without optimizing the representation format. The traffic signs are serialized by a Java *ObjectOutputStream* which also serializes the structure of the class. Hence, the first sign takes 382 bytes while each additional traffic sign consumed 83 additional bytes. Hence, it was possible to transmit at most 20 traffic signs at once for a maximum message size of 2364 bytes and a GeoNetworking overhead of 322 bytes is assumed.

5.6. Summary

In this chapter, we proposed and evaluated two versions of a novel anonymous authenticated key agreement protocol which combines ECIES with ring signatures. The protocol enables two IVSs to authenticate each other and agree on a symmetric encryption key to exchange confidential data without leaking their identity. Therefore, the protocol protects the privacy of communication and identity of the involved IVSs.

We first detailed the single steps of both the plain and encrypted version of the protocol. Then, we wrapped the messages of the protocol into GeoNetworking packets, which is a common means in VANETs to address other IVSs.

Furthermore, we analyzed which sensitive information different types of attackers might gain in terms of possible identities or applied ring size. The results show that even a passive attacker without any access to valid pseudonyms might get all information when attacking the plain version of the protocol. If the encrypted version of the protocol is applied, only an active attacker with at least access to pseudonyms applied in safety communication is necessary to get possible identities of the communicating IVS. Even if the attacker gets the possible identities, she is not able to get any attributes of the IVS besides the ones which are anyway necessary to obtain pseudonyms for the application.

In order to assess the privacy impact of the protocol we set up a realistic simulation scenario. The results show that the anonymity of the IVSs increases significantly when a larger ring size or good ring building strategy is applied. An attacker cannot simply link different executions of the protocol by the same IVS and identify often visited locations. Therefore, the protocol protects the privacy of identity and location of the involved IVSs. We also demonstrated that the number of pseudonyms each IVS applies at the same time and the duration of the attack have a clear influence on the anonymity level of the IVSs. In comparison to safety-related communication, less pseudonyms for each IVS over time are necessary to maintain a high level

of anonymity, because they can be reused without the risk of being linked by an attacker. Therefore, the amount of pseudonyms, which have to be loaded onto the IVSs, is significantly reduced. This saves storage space and communication overhead and thus helps to reduce costs. Because only pseudonyms for the particular application are exploited, the IVS does not reveal its association to other applications by executing the protocol.

In addition to the simulation, we evaluated the protocol in a real-world test setup by implementing it into real vehicles. In these tests we demonstrated that the simulated ring sizes are realistic and could be even increased when compressing the pseudonyms applied to create the ring signature. One of the main results of this real-world evaluation consists in the fact that the complete protocol can be executed in a feasible time under harsh conditions. Even two IVSs featuring a relative speed of 200 km/h and communication range of only 100 meters are able to complete the protocol for a ring size of 13 members while in communication range. In addition, the relative speed of the IVSs has no measurable influence on the overall execution time. Another interesting result was that the more privacy preserving encrypted version of the protocol has a smaller message size and no measurable influence on the total execution time, even though more data is encrypted. Additionally, we successfully evaluated the prototype implementation regarding its robustness to faults in the exchanged messages. Finally, the protocol was successfully executed by three entities at the same time and applied to exchange real payload between the IVSs. The overall execution time of the protocol is dominated by the requirements of security related operations. This can be significantly reduced by optimizing the creation and verification of the ring signature.

The novelty of the protocol proposed in this chapter consists in hiding the identity of an IVS for multiple executions while keeping at the same time the number of necessary pseudonyms smaller in comparison to existing protocols. We achieve this by combining different security mechanisms. Therefore, the protocol is more general than the single applied security mechanism and features new properties. It is also not limited to IVSs because any two identities aiming for a protocol featuring these properties can use the outlined scheme.

More and more OEMs are offering over the air updates for software running on their vehicles [NL08]. However, the distribution of software updates via mobile networks is expensive and IRS networks are not likely to be deployed comprehensively. Therefore, this protocol can be used to save monetary costs for transmission. When exploiting the scheme, the distributing ICS needs to send the update to only a fraction of all IVSs. These IVSs then distribute the update free of charge via DSRC and the proposed protocol to nearby vehicles [LHH08].

The proposed protocol is not restricted to VANET use cases, but it is especially well-suited for VANETs, because in such a context it is both expensive and some-

times rather difficult to obtain new pseudonyms. In addition, the well-known pseudonyms for safety-related communication can be reused by binding them to specific applications as shown in Chapter 4

Hence, we demonstrated that the protocol enhances the privacy of the IVSs even against active attackers with access to valid pseudonyms. At the same time, it reduces the amount of necessary pseudonyms. Furthermore, we showed that the protocol can be executed in a feasible time on real vehicles.

6 Anonymous Data Reporting

Modern vehicles are equipped with a wide variety of sensors such as wheel speed sensors, Global Positioning System (GPS) receivers, acceleration sensors, radars, lasers, or cameras. The readings of these sensors are not only of interest for the vehicle and driver, but for many other parties too.

An ICS which offers information about traffic or road conditions or up to date map data might for example be interested in these sensor readings. However, if an IVS just uploads its sensor readings to an ICS sensitive information might be derived from this data. We show in the sequel that it is for example possible to detect if an IVS was driving too fast whenever the data was tagged with a geographic position and time.

To obscure the originating IVS of the reported data and authenticate the communicating entities we apply the attribute-based ATs and anonymous authenticated key agreement between two IVSs presented in the previous chapters. Therefore, the meaning of anonymity in this chapter is that it is difficult to extract the originating IVS of the sensor data. To hide the identity of the originating IVS the sensor data is exchanged between different IVSs exploiting DSRC prior to sending them to the ICS. By way of exchanging the sensor data it is no longer possible to extract the origin and therefore create movement profiles or identify speeding IVSs. A similar scheme has been proposed for smart phones in [CGR+11].

Therefore, we show in this chapter how sensible data might be easily extracted from reported data, outline our assumptions and the considered attacker model within the application scenario, detail the proposed concept to hide the origin of the data, and propose two different strategies on how to exchange the data between vehicles before sending them to an ICS. For assessment purposes, we compare their efficiency by the means of simulation and evaluate their results. Furthermore, we give a recommendation on a good exchange strategy. This chapter is based on the paper [BH15a] and was mainly extended by its motivation and evaluation.

6.1. Motivation

An operator of an ICS could for example be interested in the data generated by the accelerometer and GPS in order to detect the size and position of potholes [EGH+08], or the information from wheel sensors in order to detect icy roads. Finally, data about traffic signs or other data collected by the built-in camera could be applied to increase the quality of street maps data for automated driving purposes [HER16]. Vehicles collecting these sensor readings use in general cellular networks or C2I communication in order to forward the data to an ICS. This ICS then processes and possibly aggregates the data. Afterwards the data is sent in the context of a subscription to other IVSs present in the surrounding in order to improve road safety. Besides sending the data to IVSs, the operator of the ICS could also sell the data to other interested parties, e.g., the government, to improve the coordination of road works, or third-party ICSs which further process the data and possibly merge them with data from other sources.

However, one should be aware of the fact that all data sent to the ICS is associated with an individual geographical position, since it is only valid in an area of just a few meters around the detection point. Therefore, the ICS is aware of the locations visited by an IVS over time and is thus able to create a movement profile of the IVS. This is a very critical privacy threat, because position data of IVSs reveal privacy-sensitive locations such as home, workplace, church or hospital. So, movement profiles are well-suited to identify the person who drives the IVS [Kru07].

Furthermore, vehicles are driving along streets. Whenever an IVS uploads at least two sensor readings with an associated location, it is possible to reconstruct the route of the IVS. Therefore, it is also possible to calculate the traveled distance of the IVS. If furthermore the timespan between the two sensor readings is known, it is possible to calculate the average speed of the vehicle between the two sensor readings with Equation 6.1. With additional data like the maximum allowed speed on the taken route, it is possible to detect if the vehicle was speeding.

$$speed = \frac{distance}{time} \tag{6.1}$$

The reconstruction of the traveled path based on uploaded sensor data has been implemented based on OpenStreetMap (OSM) data as bachelor thesis [Lae15]. First the positions of the sensor readings have to be mapped to the streets. Afterwards the route between the two points is calculated. From the route information the distance between the sensor readings was calculated and together with the timespan inserted in Equation 6.1 in order to get the average speed of the vehicle.

An example where three traffic signs, limiting the maximum allowed speed to 70 km/h, are detected by an IVS and sent to an ICS is given in Figure 6.1. The blue

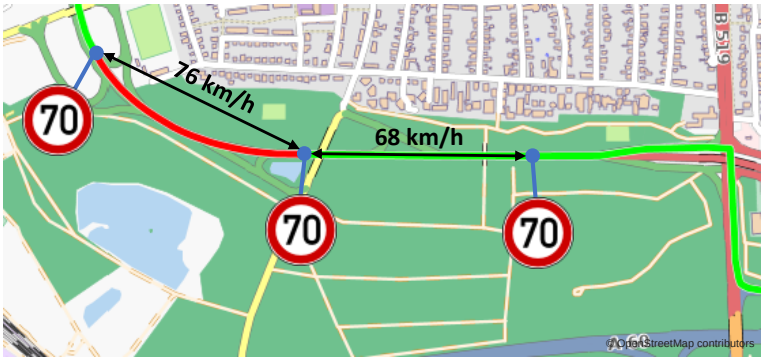


Figure 6.1.: Average Speed reconstruction example

dots are the OSM points where the traffic signs are matched to. The resulting distance of the street is 514 meter between the right and middle and 552 meter between the middle and left traffic signs. The timespan between the right and middle sign are 27 seconds and 26 seconds between the middle and left traffic sign. Subsequently, the average speed of the IVS between the right and middle traffic sign can be calculated to 68 km/h and 76 km/h between the middle and left traffic sign. In this example speed limit signs are send as data to the ICS. Consequently, it is easy to get the maximum allowed speed on this street section and determine that the driver was speeding between the middle and left traffic sign.

This shows that it is possible to calculate, based on data samples sent from an IVS to an ICS, if an IVS was speeding. This data might be exploited to harm the driver by selling it to insurance companies which might increase the insurance contribution.

Depending on the implementation it is also possible to detect if a driver is speeding by having only one data sample, if the real sensor data position, e. g., the exact position of the traffic sign, is known to the ICS. A regular GPS receiver periodically outputs the current position in a defined frequency. Common frequencies are 1, 5, and 10 Hz. Whenever an application on the IVS takes these raw positions and does not interpolate the sensor readings, it must utilize the previous position or wait for the next output of the receiver in order to tag the sensor readings with it. Depending on the speed of the IVS the recorded position might be closer or farther to the real position of the sensor event. We assume in the sequel that the application utilizes the last known position prior to the sensor event. In case the event was recorded immediately prior to the next position update, the applied position data is up to one second old. In order to calculate the maximum position aberration for a given position update

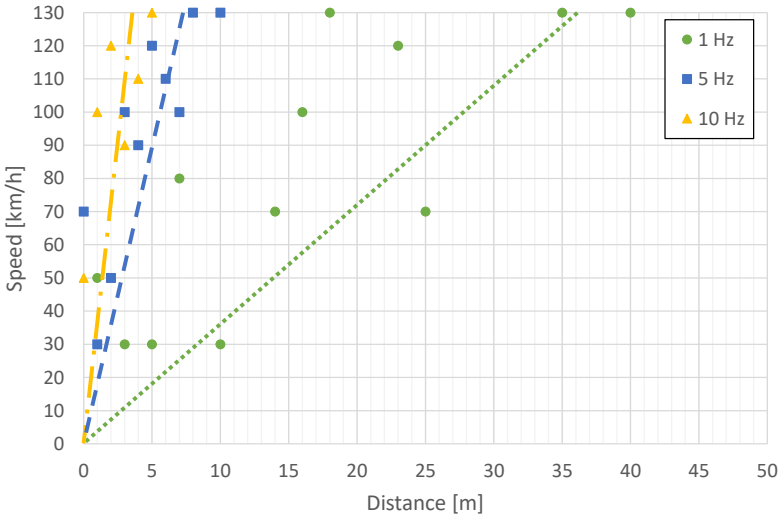


Figure 6.2.: Speeding detection by deviation measurement

frequency and vehicle speed Equation 6.2 can be utilized.

$$maxDistance = \frac{speed}{frequency} \tag{6.2}$$

If the ICS knows the maximum allowed speed at the sensor recording location and the position update frequency of the exploited GPS receiver, it is able to calculate the maximum possible aberration introduced by the speed of the IVS. If the driver is speeding, the position might be further away.

Figure 6.2 illustrates the maximum aberration for different position update frequencies dependent on the speed. The dotted green line shows the maximum aberration for a position update frequency of 1 Hz, the blue dashed line for 5 Hz and the yellow dash dot line for 10 Hz. The green circular points are recorded at 1 Hz, the blue rectangular points at 5 Hz and the yellow triangular points at 10 Hz. In all cases where the dots have a greater distance than the corresponding maximum aberration line, the driver of the IVS was speeding. However, this might also be true for some of the other points in case the tagged position was recent.

Therefore, it is possible to detect if a driver is speeding by only one sensor reading. However, if the origin of the sensor data is no longer known, it is not possible to link the data to a specific IVS or driver. An ICS could of course decide to anonymize

the data received by the IVS, but it is not able to prove the anonymization to its subscribers. In order to prevent the ICS from creating movement profiles or detecting locations where a driver was speeding, we introduce path hiding strategies for IVSs equipped with DSRC as a powerful means to enhance the privacy of the driver. While traveling along the roads an IVS usually encounters many other IVSs. With the help of DSRC, IVS create a VANET with other IVSs that allow a direct data exchange. The advocated path hiding strategies exploit DSRC in order to exchange collected sensor data between IVSs in communication range prior to contacting the ICS. In case the sensor readings are exchanged between IVSs before an upload operation, an attacker at the ICS attempting to create movement profiles is no longer able to determine the originating IVS of the sensor readings at hand. Thus, no IVS should upload its own sensor readings to the ICS directly. Moreover, instead it uploads only received sensor readings of some other IVSs in its communication range.

6.2. Application Scenario

We assume that IVSs are collecting sensor readings of interesting events by means of their built-in sensors. These sensor readings consist of the position, a coarse time, and the actual sensor value. The geographical position is necessary because the data is in general valid only in a small area. An exact timestamp is not necessary in our context, because most events are valid for a long time period. In the use case where potholes are detected, for example, it does not matter at which second, minute or hour a pothole was detected. It is sufficient to know the day or week of its detection. The same holds for the use case where traffic signs are detected.

We assume that the collected data is digitally signed and sent over cellular networks or C2I communication to an ICS. The digital signature is necessary to ensure the authenticity and integrity of the data. The ICS first aggregates the data. Afterwards, the ICS extracts the interesting information and distributes it to subscribed IVSs in the region of interest or to other ICSs.

6.2.1. Attacker Model

The considered attacker has access to all data received by the ICS. Due to the digital signature of the sensor data, the attacker knows which data was uploaded by the same IVS. Even if the IVSs regularly change their identity, for signing the sensor data, it is still possible for an attacker to reconstruct the relation between the sensor data and a specific IVS [WMKP10]. Hence, the attacker is able to take the location of the sensor readings to reconstruct the paths of the IVSs and thus to get information about, e.g., the workplace or home of the drivers.

To prevent the reconstruction of such privacy-sensitive information in the first place, IVSs should exchange their sensor readings with other IVSs in the VANET. A non-safety channel should be used not to impair any safety applications. By means of exchanging sensor readings before the upload operation, the attacker at the ICS is no longer able to decide on the origin of the sensor readings. An IVS does not send its own sensor data to the ICS directly, it uploads only the data collected by other IVSs.

This mechanism, however, introduces new possible attacks. An adversary on the road could, for example, record the data transmitted between two IVSs and offer it by its own to other ICSs without doing the work of collecting it. An adversary could achieve this by getting control over some nodes in the VANET. To make things worse, an adversary could not only record but also send false messages to participating IVSs. This can lead to forwarding false sensor readings to the ICS. Therefore, the protocol aimed to exchange data needs to send the data confidentially. Protocols exploiting ECIES like the one proposed in Chapter 5 are best suited to be used in VANETs to meet this requirement. To prevent the injection of messages from manipulative IVSs and establish trust between the IVSs, mechanisms like remote attestation [OYN+08] might be exploited too.

Because the newly introduced threats can be solved by existing techniques, we concentrate in the remainder of this chapter on how to confuse an attacker at an ICS by exchanging the sensor readings between IVSs before uploading them.

6.3. Scenario

An example for a sensor data exchange scenario is depicted in Figure 6.3. It shows roads connecting a school, hospital, bank, and cinema. The sensor values Sn of interesting events are recorded by passing vehicles. Vehicle A is aiming from the hospital to the bank, while vehicle B takes the way from the school to the cinema (solid lines). If A and B upload their collected sensor readings directly, the ICS can easily reconstruct the paths of both vehicles.

In order to enhance the privacy of its drivers, the IVS exchange the collected sensor values, when they are in proximity. In the example of Figure 6.3 this happens in the area limited by the dotted circle. Up before reaching this area, A already collected the sensor data of $S1$, $S2$, and $S3$. Vehicle B collected the sensor data of $S5$. After the data exchange both vehicles continue cruising. A will now upload $S5$ received from B near the school and the self-detected sensor value $S6$ near the bank. The uploaded sensor values by B are the three received from A ($S1$, $S2$, and $S3$) and the self-collected $S4$ near the cinema.

If an attacker at the ICS now tries to reconstruct the paths of these IVSs, the data

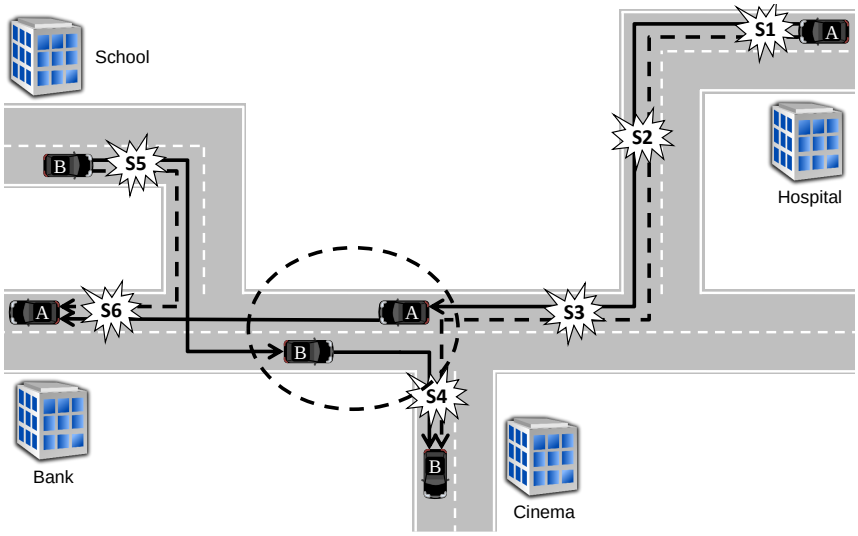


Figure 6.3.: Example for a sensor data exchange

will suggest that A was driving from the school to the bank, while B was driving from the hospital to the cinema (dotted lines in Figure 6.3). It depends on the applied strategy, e.g., when to exchange the data with other vehicles and when to upload it, how much an attacker at the ICS can be confused. In the following we propose possible strategies on how to exchange the collected data between vehicles in order to enhance privacy.

6.4. Proposed Strategies

The different stakeholders have conflicting requirements on the data exchange. In general, the ICS requires the data as soon as possible, while in contrast the drivers are interested in better privacy. Privacy in this context means that the position information associated with the uploaded sensor data is far away from the own route of the IVS and were previously collected by as many other IVS as possible, which is rather time consuming. Another requirement by the ICS is as less load as possible at the server to process the data, while in contrast the IVS may aim to upload some data multiple times in order to confuse a possible attacker. We address these conflicts in the evaluation detailed in Section 6.6.

We suggest two different strategies which are, in comparison to the ones proposed

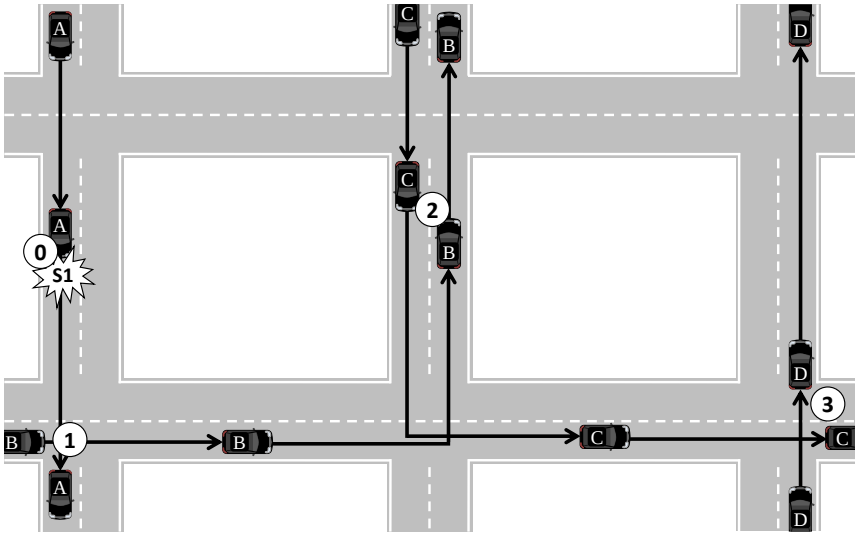


Figure 6.4.: Example of the exchange based strategy

in [CGR+11], optimized for VANETs. The first one is based on the number of exchanges. If the sensor data has been exchanged n times, it will be sent to the ICS. The second strategy is based on the distance between the position of the sensor reading and the position of the own IVS. If the sensor readings collected by other IVS have at least the distance d from the own route, then the data is uploaded to the ICS. In the following we discuss these strategies in detail.

6.4.1. Exchange Based

When applying the *exchange-based strategy*, an IVS decides from the number of times a sensor reading has already been exchanged, if it has to be uploaded to the ICS or sent to another IVS. For this strategy a global parameter n is necessary, which defines the number of times sensor readings have to be exchanged between IVSs before uploading them. The idea of this strategy is that the more often the sensor readings are exchanged, the more they are mixed with those of other IVSs.

An example for this strategy employing 3 exchanges is given in Figure 6.4. Vehicle A records the sensor value $S1$ at position 0. Later it meets vehicle B at location 1, where $S1$ is exchanged for the first time. B now carries $S1$ until it meets vehicle C at location 2, where $S1$ is exchanged for the second time. Vehicle C stores the sensor reading until it encounters vehicle D at location 3. After the exchange between C

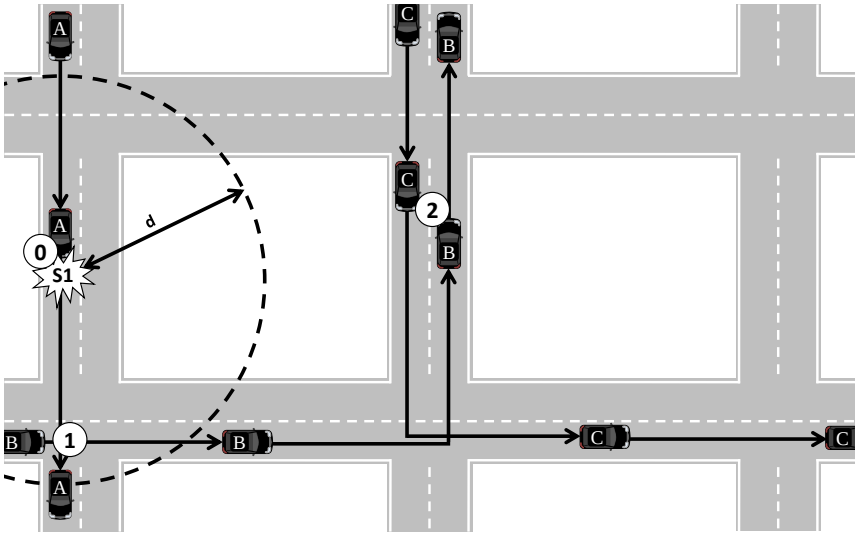


Figure 6.5.: Example of the distance based strategy

and D the sensor value $S1$ is uploaded by D, because it has already been exchanged 3 times.

6.4.2. Distance Based

An IVS applying the *distance-based strategy* calculates for each received sensor value the distance to its own route. If this distance is greater than a certain value, the IVS uploads the data. Otherwise it exchanges it with another IVS. For this strategy a parameter d is necessary, which defines the minimum distance between the route of the own IVS and the location of the uploaded sensor data. Therefore, this strategy ensures that all uploaded sensor values have a minimum distance to the own route. This strategy ensures better privacy, since all sensor readings sent to the ICS feature a minimum distance to all visited locations.

This strategy is illustrated in Figure 6.5. Vehicle A records the sensor value $S1$ at location 0 and stores it locally. When A encounters vehicle B later on, the sensor data $S1$ is exchanged. B now checks, if the distance between its own route and the position associated with $S1$ is greater than d. In the example it is smaller, so the vehicle does not upload $S1$. At location 2 B encounters vehicle C and sends the sensor data $S1$. C now checks the distance criterion as well. Because the distance is greater, C uploads the sensor data $S1$ to the ICS.

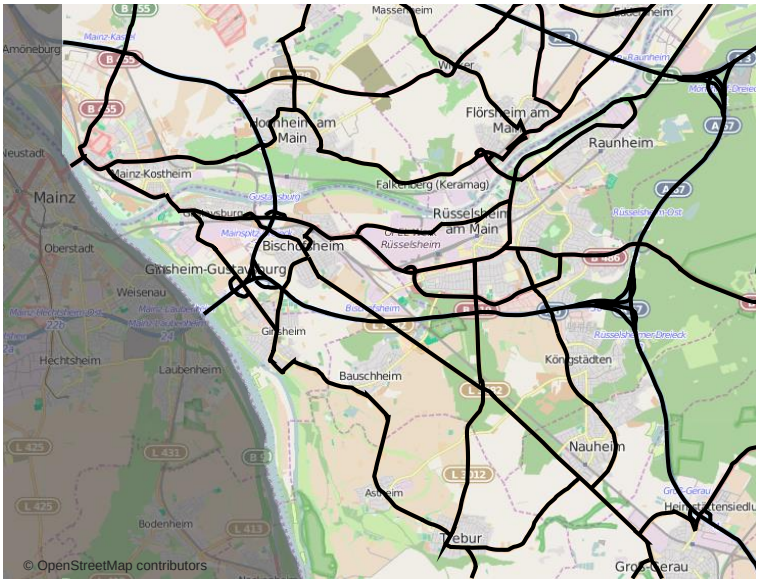


Figure 6.6.: Simulated street network

6.4.3. Orthogonal Parameters

Both strategies depend on the parameters *TimeBetweenExchanges* and *DataDuplication* which are independent from the used strategy. *TimeBetweenExchanges* denotes the time an IVS waits between two exchanges. Otherwise, the originating IVS may possibly get its own sensor readings back and upload it. For example, consider three IVSs (A, B and C) being in the communication range of each other. First, A exchanges its sensor readings with B. B now immediately exchanges the received readings with C, which now will send the data back to A, which is the actual originator of the sensor readings. To suppress such unwanted behavior, we use *TimeBetweenExchanges* to introduce a delay between two exchanges.

The parameter *DataDuplication* defines the number of times the detecting IVS sends the same sensor readings to other IVSs. This parameter is applied to confuse the attacker, because it actually gets the same sensor data several times from different IVSs without knowing the origin anymore.

6.5. Simulation Scenario

For the deterministic simulation we exploited VSimRTI [Sch11]. As a use case for sensor data upload we focused on the upload of speed-limit traffic signs to an ICS. The simulation scenario utilized in this chapter is based on the one developed in [Dah14]. In the scenario, all higher order road data in the area between Groß-Gerau and Mainz, Germany, was imported from OSM to be used as the geographic area. Figure 6.6 shows a map, where the imported road data is highlighted. It was observed that the most speed-limit traffic signs can be found on this type of roads. Therefore, the precise position information of all speed-limit traffic signs on these roads throughout the simulation area were collected by means of trips with real vehicles and were added to the simulation. Subsequently, routes for the vehicles in the simulation were created as follows: From each road that enters the simulation area and each city within, a route to each road leaving the area and all cities within the area was defined. Then the traffic density on the routes was defined according to the annual average daily traffic as documented in [Hes10]. It is further assumed that ten percent of all vehicles considered in the simulation are equipped with the sensor application. We implemented both proposed strategies onto the vehicles and run separate simulations to evaluate them.

6.6. Evaluation

We varied the basic strategy parameters as follows: For the exchanged-based strategy we applied 1, 3, and 5 as the number of necessary exchanges n before an upload operation. When the distance-based strategy was applied, we used 1.5, 3, 4.5, and 6 km as the minimum distance d between the geographic position associated with sensor readings and the own route of the IVS. For *TimeBetweenExchanges* we applied 60, 120, 180, 240, and 300 seconds. For the parameter *DataDuplication* we used the values 1, 3, and 5. Because of the huge simulation area one simulation run took about 20 hours. Therefore, we were not able to run the simulations with different route choices or varying random seeds.

6.6.1. Metrics

For the evaluation of the different strategies the following metrics have been applied.

k-anonymity

Describes the number of different IVSs whose sensor readings a certain IVS uploads. If an IVS sends data of many different IVSs to the ICS, it is more difficult to get the

identity of a single IVS. Therefore, a higher value of k-anonymity is better.

Spatial obfuscation

Defines the mean distance between the position of the uploaded sensor data to the route of the IVS. For an IVS it is better to have a larger distance between the uploaded sensor values and the own route. Then it is more difficult for an attacker to reconstruct both the path and the sensitive locations of an IVS.

Upload delay

Describes the time elapsed between the collection of the sensor reading and the point in time when it is received by the ICS. The ICS wants this value as small as possible, i.e., up-to-date data.

Additional channel load

Describes how many times the IVS in total exchange sensor data between each other using C2C communication. This metric is necessary to estimate the additional channel load caused by the exchange operations.

Amount of uploaded data

Defines the number of sensor readings uploaded to the ICS. The more additional sensor data is received by the ICS, the more computational effort is necessary to process the data. Therefore, the ICS aims to receive each sensor reading only once, while the IVSs might duplicate some sensor data in order to increase their privacy.

Storage

Defines how much storage space on the IVS is necessary to store the sensor readings. Storage is necessary to buffer the data while waiting for the next exchange with another IVS or for a connection to the ICS.

6.6.2. Results

In this section we present only the simulation results where the *DataDuplication* parameter is set to 3. For other values the results are similar except for the *amount of uploaded data* metric, which depends on it. The results for other values of *DataDuplication* are given in Appendix A.2.

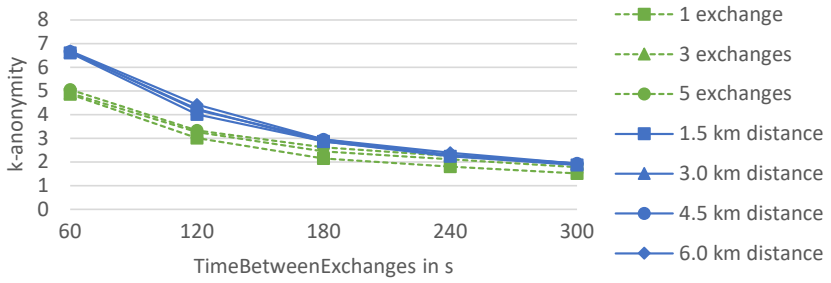


Figure 6.7.: k-anonymity as a function of *TimeBetweenExchanges*

k-anonymity

Figure 6.7 shows that the k-anonymity value of the IVSs decreases for both strategies with *TimeBetweenExchanges*. If it is higher, there are less exchanges between IVSs. Therefore, an IVS receives data from fewer other participants. The strategy-dependent parameters have no significant influence on the k-anonymity of the IVSs. Please note that the k-anonymity for the proposed strategies is quite similar. However, it tends to be slightly better for the distance-based strategy and small *TimeBetweenExchanges* values.

Spatial obfuscation

The spatial obfuscation increases with *TimeBetweenExchanges* for both strategies (see Figure 6.8). If IVSs wait longer between two exchanges, then they also travel further in this time. Therefore, the sensor readings are also carried further from their origin on average.

For the exchange-based strategy the spatial obfuscation increases with the number of exchanges. Each time the data is exchanged, it is further carried away from its origin. For the distance-based strategy it also increases with the minimum distance d. When the minimum value is increased, it automatically increases the mean value.

For the smallest simulated value of the distance-based strategy the evaluated distance is higher than for all simulated number of exchanges of the exchange-based strategy.

Upload delay

This value increases for the exchange-based strategy linearly with *TimeBetweenExchanges* (see Figure 6.9). When the IVSs wait longer between two exchanges, the

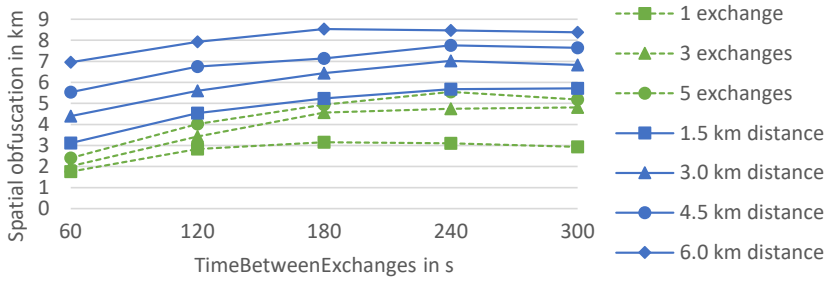


Figure 6.8.: Spatial obfuscation as a function of *TimeBetweenExchanges*

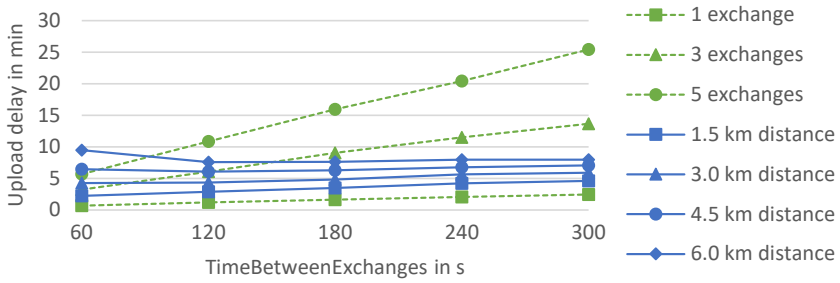


Figure 6.9.: Upload delay as a function of *TimeBetweenExchanges*

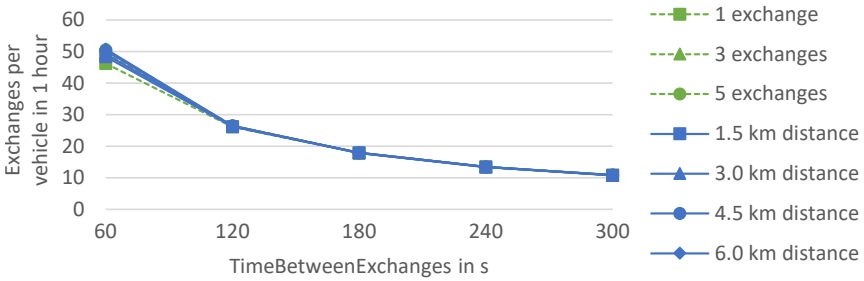


Figure 6.10.: The number of exchanges per vehicle in one hour as a function of *TimeBetweenExchanges*

sensor data also needs longer to reach the desired number of exchanges.

For the distance-based strategy the upload delay stays nearly constant. The speed the IVS move away from the sensor data position is independent of *TimeBetweenExchanges*. The delay for the distance-based strategy is for all simulated distances between 2 and 7 minutes, while for the exchanged-based strategy it increases to more than 11 minutes for 3 exchanges and to more than 20 minutes for 5 exchanges.

Safety relevant sensor data like an icy road can be sent in addition to all IVSs in proximity via DSRC. In this way it is ensured that other nearby IVSs receive this information without the introduced delay in time. The messages which are received by the ICS with an additional delay may then also sent to other IVSs located outside of the DSRC communication range. Therefore, these IVSs can earlier prepare for the hazardous location.

Additional channel load

This value is almost equal for both strategy and decreases with increasing *TimeBetweenExchanges* values. The IVSs wait longer between two exchanges and therefore can execute less exchanges per time unit. Because the IVSs detect a speed-limit traffic sign on average every 93 seconds, they almost execute the protocol immediately after the waiting period is over. The number of exchanges for a single IVS per hour are shown in Figure 6.10.

Amount of uploaded data

The simulation results show that the amount of uploaded data is only influenced by the parameter *DataDuplication*. This parameter is the only one that actually puts

additional data into circulation. The amount of uploaded data increases linearly with *DataDuplication*.

Storage

The maximum number of sensor data to store at the same time was in all simulation runs between 14 and 240, which results in less than one Megabyte of storage. This is no issue, because todays vehicles carry Gigabytes of data for, e.g., navigation.

6.6.3. Recommendations

The *spatial obfuscation* and *k-anonymity* is higher for the distance-based strategy. The distance-based strategy also has a lower *upload delay*, while both strategies generate a similar amount of *additional channel load*.

Therefore, we recommend to use the distance-based strategy, since it ensures a higher privacy of the IVS. In addition, it uploads the data faster than the exchange-based strategy. We recommend also a high value of *TimeBetweenExchanges*, because it increases the privacy of the IVSs, reduces the additional C2C channel load and increases the upload delay only slightly.

6.6.4. Real Hardware

Besides the simulation we also considered the prototype implementation of the exchange protocol outlined in Chapter 6 in order to evaluate the path hiding strategies. We ran it on a vehicle which was in addition equipped with a piece of software to detect and store passing traffic signs [Bre14]. With this setup we successfully exchanged the detected traffic signs with other vehicles. We implemented the exchange based strategy with a parameter to set the number of exchanges prior to sending the traffic signs to an ICS. In different evaluation runs with different numbers of necessary exchanges prior to sending the traffic signs to the ICS they were always exchanged according to the parameter.

6.7. Summary

In this chapter we proposed a scheme to hide the origin of sensor data sent from an IVS to an ICS. The scheme utilizes DSRC to exchange the sensor data between IVSs in a VANET prior to sending them to the ICS. As protocol the one outlined in Chapter 5 was exploited to ensure the communication privacy of the participating IVSs. We outlined two different strategies on how to exchange the data between the IVSs before uploading it. One is based on the distance between the uploaded data

and the own route while the other is based on the number of times the sensor data has been exchanged.

We created a simulation scenario to evaluate the different strategies. This scenario contained sensor data collected by real vehicles, a real vehicle density and a real road network imported from OSM.

We showed by the presented simulation results that exchanging sensor readings with other IVSs obstructs the creation of movement profiles at an ICS. The data reported to the ICS by an IVS belongs to many other IVSs too. Furthermore, it is several kilometers away from the own route, while the introduced time delay for uploading the data is only a few minutes. The scheme also protects the privacy of location and identity of the originating IVS.

Compared to previous work we show that the mechanism of exchanging sensor data prior uploading it to a central server is also well-suited to protect the privacy of IVSs within a VANET. To archive this, we optimized strategies for exchanging sensor data between IVSs and built up a realistic evaluation scenario in a simulator to evaluate them. Furthermore, we outline threats which result from the special characteristics of IVSs.

The proposed strategies may further be improved by refraining from exchanging all collected sensor readings at once. Instead they could be divided into smaller blocks. It is also possible to mix the outlined strategies. The distance-based strategy may be, for example, extended by uploading the sensor values not only after a certain distance, but also after n exchanges. Another modification would be to upload the data after a certain time period even if it is not yet exchanged n times or the distance is smaller than d.

To authenticate the uploading IVS, all data send to the ICS is digitally signed by previously obtained attribute-based ATs for the utilized application. Therefore, the ICS is aware of the AT applied by the uploading IVS. Ring signatures might be considered to prevent this. In Chapter 5 we showed that an attacker is not able to resolve the identity of the IVS if ring signatures are applied. Instead the attacker only gets a set of possible signers part of the application. Furthermore, an attacker at the IVS does not learn any attributes besides the necessary ones to obtain the AT.

When sensor readings are exchanged between IVSs prior to sending them to an ICS, an attacker at the ICS is no longer able to identify the originating IVS of sensor data. However, it might be possible to identify the vehicle model by the sensor reading itself. For example, cameras in some vehicles might always detect the same wrong traffic sign at a certain location. An attacker might use machine learning algorithms in order to determine the vehicle model from this information. As countermeasure, an IVS might exploit the information the machine learning algorithm applies in order to determine the vehicle model. It could change the sensor data information prior to the upload accordingly. Then an attacker at the ICS is no longer

able to identify the model of the detecting vehicle.

7 | Anonymous Geocast

In the previous chapters we presented mechanisms for IVSs to authenticate for specific applications based on attributes, authenticate each other anonymously and agree on a symmetric encryption key at the same time, and hide the origin of sensor data sent to an ICS.

Most information in ITS is only relevant in a certain geographic area. Therefore, the distribution of data in a defined geographic area is a common use case for ITS applications. An ICS may have for example received sensor data from IVSs which is relevant for all other IVSs in the region. A suitable transport mechanism is necessary to distribute the data to the subscribers of the application located in the target region. Different methods to distribute data in a certain geographic area exist. However, as discussed in Section 3.4, none of them is well suited for ITS applications.

Therefore, we propose an anonymous geocast scheme in this chapter, which allows an ICS to distribute messages to all IVSs subscribed to its application and located within a certain geographic region. To protect the privacy of IVSs, our distribution mechanism does not store information about the receiving IVSs location or subscribed applications. We compare it with state of the art solutions and, in addition, provide a prototype implementation which has been evaluated on real DSRC hardware. The remainder of this chapter is a polished and extended version of the publications [BH16a] and [BH16c]. We furthermore filed this scheme as patent [BH15b].

7.1. Motivation

Applications in ITS with content such as map data update and information or warnings on weather hazards, wrong way drivers, traffic jams, or road works ahead require that the corresponding messages are distributed in a specific geographic region, called *dissemination area*. A mechanism that spreads messages in a certain geographic region is called *geocast* [NI97].

If an IVS acts as the origin of such a message, it is typically already located within the dissemination area, because generated messages are based on locally detected or triggered events. Therefore, it simply distributes the message to all IVSs in communication range which are part of the VANET directly exploiting DSRC. The receivers continue to forward the message in the dissemination area by means of suitable routing algorithms [Mai04].

Besides of an IVS, an ICS may also be the origin of such messages. An ICS can be in this case, for example, a Traffic Center or a Service Center operated by an OEM. It may also have access to a meteorological service or to a database of up-to-date road works information to generate the messages. Typically, an ICS is stationary and accesses the ITS network via the Internet, i.e., it is not integrated into geographic wireless routing mechanisms. Therefore, the messages have to be transported to an edge of the geographic wireless routing network in the target region first. The final transmission medium can be part of any wireless communication technology, most probably mobile networks or DSRC. The IVSs located in the area can then further spread these messages.

There exist further requirements for a geocast such as receiver subset selection, data encryption, dynamic overlapping dissemination areas, and message validity periods. Various ITS applications are addressing a subset of the local IVSs only, for example paid application subscriptions or all IVSs of a vehicle brand. Hence, application data delivery must be restrictable to an IVS target group. Messages of some applications may further be of a commercial value and therefore require an appropriate protection. Given that an IVS does not have a high computational power onboard, it makes sense that only the desired IVSs receive and process these messages. The dissemination area of the messages may also be dynamic. An ICS could, for example, warn about distinct weather hazards present in various areas at the same time. The dissemination area may in addition change over time. Another special requirement of messages sent by ITS applications is that they may feature a strict validity period, in which they should be sent to each IVS entering the dissemination area too.

LTE is the current high speed communication standard for mobile networks. Several LTE-based ITS geocast approaches have meanwhile been proposed. However, none of them satisfies the requirements of the outlined ITS applications because they either do not scale with the number of recipients or they are not able to automatically distribute messages to IVSs entering the dissemination area. Furthermore, the procedure becomes rather complicated when different messages have to be distributed in various frequently changing geographic areas at the same time. In addition, none of them considers the privacy of the IVSs accordingly. Some hurt the location privacy by tracking the IVSs in order to determine the ones present in the target area. This might also reveal the identity of the IVSs. Furthermore, some also require the information about all exploited applications of an IVS. When exploiting DSRC, IRSs

within the dissemination area are in principle able to distribute the messages to relevant IVSs. However, DSRC currently does not support an addressing of a group of IVSs as the only receiver of a message. Therefore, the current mechanisms for mobile networks and DSRC are not suitable to distribute ITS messages to a group of IVSs in a given geographic area.

One has to consider the fact that the messages need to be distributed by different means because it will be rather common that there will be IVSs only equipped with either a mobile networks connection or DSRC. Furthermore, none of the communication technologies is perfectly suited to the envisaged application. The coverage of DSRC is limited because IRS Networks are unlikely to be deployed comprehensively. However, they will be most likely deployed at dangerous locations. In contrast, mobile Networks are widely available and provide a nearly complete coverage, but there is a fee to transmit data over the network. DSRC does not charge a transmission fee, only the costs for setup and operation. Therefore, a new mechanism is necessary to distribute messages of ITS applications efficiently in a certain geographical area, preferable via different communication technologies like mobile networks and DSRC.

These are the main reasons why we propose AGfIA, which enables an ICS to forward an ITS message to all IVSs subscribed to its application and located in or entering a certain area. It can handle the distribution of different messages at various, even overlapping, areas at the same time. The proposed scheme works for a versatile distribution via both technologies, i.e., mobile network and DSRC. Moreover, it protects the privacy of the IVSs, whereas no central entity is able to track the exploited applications of an IVS. This is achieved by minimizing the information on current application subscriptions of an IVS being stored within the network. Furthermore, no central entity tracking the locations of the IVSs is introduced.

7.2. Requirements

Different kinds of ITS applications require a geocast mechanism to distribute messages. To support a wide variety of such applications, a geocast mechanism for ITS applications must fulfill different requirements. In the sequel we discuss these requirements in detail.

Multiple Applications: A geocast mechanism in general introduces overhead. In order to minimize it, such a mechanism should be generic enough to be exploited by quite different applications. This way, applications do not have to maintain a geocast themselves. The overhead is bundled to a single mechanism and thus minimized.

Receiver Groups: Usually an IVS does not employ all available applications. Each IVS does subscribe to the applications it is interested in. In order to transmit the messages only to the subscribers of an application, a geocast mechanism should support the addressing of a subset of all IVSs present in a given area.

Dissemination Area: Each ITS geocast message has a dedicated dissemination area. Some applications, like a weather hazard warning, might intend to distribute messages in a large area like a whole state, while others, like a particulate matter emission or road works warning, target only a town or just a road section. The dissemination area can also change over time, for example for moving road works. An application might further distribute different messages to distinct areas at the same time. In addition, these areas may overlap. Therefore, a geocast mechanism should support individual dissemination areas for each message.

Validity Period: Events in ITS are usually temporary. To avoid dedicated cancellation messages, distributes messages should have a validity period. This period may last several minutes for, e. g., traffic jam ahead warnings, hours for, e. g., weather hazard warnings, or even days for, e. g., road works warnings. During this validity period the ITS message should be transmitted to each IVS entering the dissemination area. Accordingly, a geocast mechanism needs to support the transmission of messages to all IVSs entering the dissemination area.

Content Type: ITS messages are small and thus need to be handled differently compared to large data streams. Therefore, a geocast mechanism does not need to support the transmission of a huge amount of data but it should be optimized for small messages. This simplifies flow management and buffer design of involved entities and reduces the system complexity.

Scalability: An ITS application may be exercised by quite many IVSs. Consequently, a geocast mechanism should be able to scale for a large number of receivers.

End-to-End Delay: Some ITS applications like a wrong way driver warning require a real-time delivery of the messages in the dissemination area. Otherwise, the validity period of urgent messages may be expired when the messages arrive at the receivers. Therefore, a geocast mechanism should have an as small as possible end-to-end delay.

Efficient Transmission: In order to avoid unnecessary load within the communication network at hand, ITS messages should be transmitted in an efficient way. This covers mechanism and message overhead as well as message duplication.

7.3. Scheme

ITS applications, like a weather hazard warning, require a geocast to distribute relevant messages to all IVSs located in a specific geographic area. As communication technologies we consider LTE for mobile networks and IRS Networks for DSRC. In LTE, the clients are connected to an eNodeB which is linked to the core network of the MNO. We assume a Mobile Network Central Station (MN CS) as part of the core network to handle all incoming ITS geocast messages. An IRS Network consists of one or multiple IRSs, which are connected to one IRS Central Station (IRS CS). Like the MN CS for mobile networks the IRS CS handles all incoming ITS geocast messages. In case that the network consists of only one IRS, the IRS CS may also be part of this IRS. IVSs communicate with the IRS Network if they are in its communication range. Both, the MN CS and the IRS CS, are connected to the Internet.

AGfIA can be exploited for mobile networks like UMTS and LTE as well as for IRS Networks. The mechanisms to register for ITS geocast messages as well as the way how the messages are distributed are described for LTE and IRS Networks in the sequel. Furthermore, a suitable message format for both communication technologies, a possible usage-based billing mechanism, and an example are detailed. We show that our novel mechanism firstly satisfies all requirements outlined in Section 7.2 and secondly protects the privacy of each IVS.

7.3.1. IVS Registration

To initiate the geocast mechanism, each IVS has to register at the central station of the network operator first. This registration is independent from the exploited ITS applications. For the two network types different mechanisms need to be applied. They are detailed as follows.

LTE

Registering for geocast messages in LTE is similar to joining an MBMS User Service [3GP13b]. There devices join MBMS User Services in order to receive messages belonging to these applications. In comparison to MBMS not only one application utilizes this User Service. Instead, all ITS applications are handled by the same MBMS User Service. Therefore, each IVS has to join only one User Service, independent of the number and types of ITS applications it runs. This protects the

privacy of the IVSs because the MNO does not learn the applications an IVS exploits. Therefore, the subscribed IVSs of an application remain anonymous to the MNO.

Furthermore, we assume a continuous MBMS service to minimize the time overhead for setting up an MBMS session. A time sequence diagram detailing the registration scheme is depicted in the upper part of Figure 7.2.

IRS

For IRS Networks no registration is necessary because of the ad-hoc characteristic of the network. As a consequence, the operator of an IRS Network is not able to track the IVSs subscribed to a certain ITS application. Therefore, the receiving IVSs stay anonymous.

7.3.2. Message Distribution

Whenever an ICS aims in distributing a message in a geographic area, it has to look up the present mobile and IRS Networks in the area first. To find relevant networks, the ICS requires a coverage map of all mobile and IRS Networks it has a contract with. We assume that there exists an ICS which provides this information [PW12]. After relevant networks have been identified, the ICS passes the message to the central station of each network, including meta-information about the dissemination area, a distribution frequency, an expiry time, and an AID. The dissemination area defines the region in which the message shall be spread. To distribute the message also to IVSs entering the dissemination area, it is repeated at the given distribution frequency until the expiry time. The AID uniquely identifies an ITS application responsible for the message. Therefore, it is necessary for an IVS in order to determine if the particular message is relevant or not. The message distribution by the different network types as discussed in the sequel is illustrated in Figure 7.1 and the corresponding sequence diagram is presented in the lower part of Figure 7.2.

LTE

Each time an ICS aims at distributing an ITS message in a certain area, it passes the messages to the MN CS. Its functionality is similar to the BM-SC and MBMS-GW in MBMS and can therefore be integrated to these entities. It first identifies the relevant eNodeBs to distribute the message. This is done with the help of a database, containing the position, communication range, and address of each eNodeB within the network. Then, the geocast message is forwarded to these eNodeBs by means of Xcast [BFI+07]. Upon reception the eNodeB stores the message locally. All locally stored messages are then sent at the given frequency to all IVSs in the communication

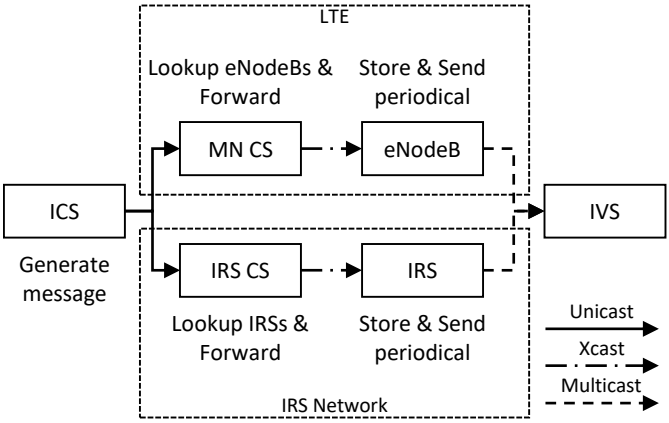


Figure 7.1.: Message distribution in AGfIA

range belonging to the ITS MBMS User Service until they expire. Furthermore, the eNodeB is aware of all connected IVSs which are member of the User Service for ITS applications. Therefore, it could also send a unicast message to this new IVS immediately after connecting instead of periodically repeating the message on the broadcast medium to all IVSs. This might be even more efficient [SZ11].

This mechanism protects the privacy on subscribed applications of the IVSs since each IVS which is a member of the ITS MBMS User Service for ITS applications receives the message. Therefore, the MNO is not able to determine which IVS processes the messages and consequently cannot identify the applications exploited by an IVS. Furthermore, the IVS does not periodically send its position to a new central entity. This prevents tracking attempts.

IRS

For AGfIA on top of DSRC, the geocast messages are passed from the ICS to the IRS CS. The IRS CS has access, like the MN CS in LTE, to a database containing the position, communication range, and address of each of its IRSs in order to select the relevant ones for distribution. After selection, the messages are forwarded like in LTE via Xcast to these IRSs. There the message is stored locally and sent periodically according to the given frequency to all IVSs in communication range until it expires.

Considering that the IRS does not get any feedback which IVS in communication range processes the received message, no entity is able to determine the applications

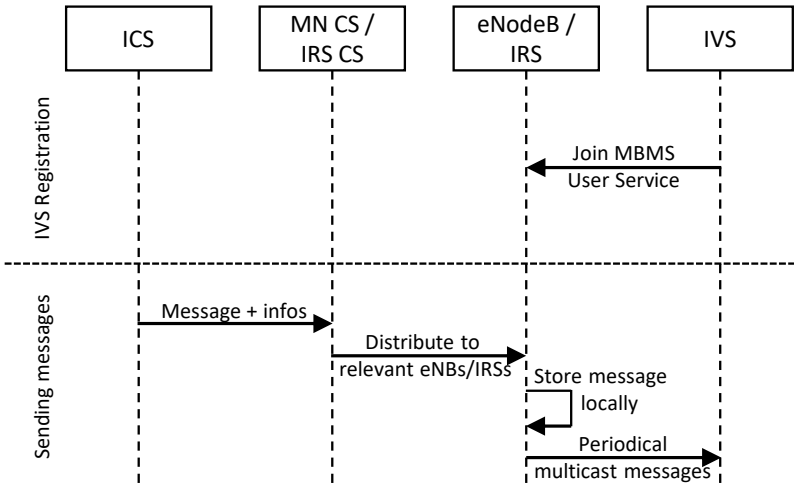


Figure 7.2.: Time sequence diagram

exploited by a certain IVS. Therefore, the distribution of geocast messages over IRS Networks also protects the privacy of the IVSs.

7.3.3. Message Format

We aim at the GeoNetworking message format as standardized in [ETS14a] and described in Section 2.3.1. This format was developed to exchange messages by means of DSRC between IVSs or between IVSs and IRSs. Therefore, it is well-suited for a geocast over an IRS Network. However, the exploited message format can be also applied for LTE.

GeoNetworking supports unicast, anycast, and broadcast messages. For the outlined scenario it is necessary to support the addressing of a group of IVSs too. Hence, we provided support for multicast messaging by adapting the GBC/Geographically-Scoped Anycast (GAC) header. All GeoNetworking messages are secured by cryptographic mechanisms in order to protect their content.

Besides of the adapted header aimed to support multicast, we apply the message format as denoted in [ETS14a] and exploit BTP [ETS14b] as the transport protocol.

The detailed format of the header supporting multicast is illustrated in Figure 7.3. The changes compared to the original GBC/GAC header are as follows. We first removed the location of the source, because it is an ICS whose location is not relevant for the receiving or forwarding IVSs. Additionally, we utilize the reserved octets

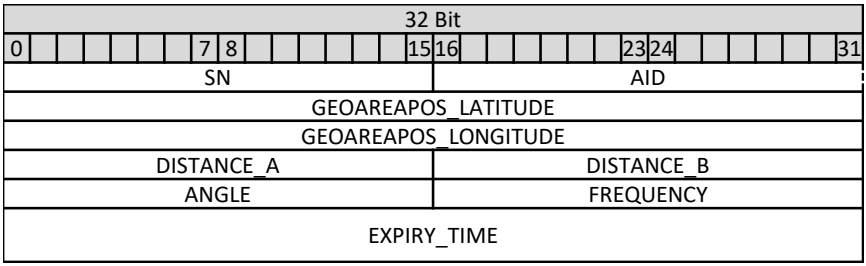


Figure 7.3.: Detailed message format of the geographically scoped multicast

to encode the AID (*AID*) into the message and add fields to embed the frequency (*FREQUENCY*) and expiryTime (*EXPIRY_TIME*) of the message. The multicast routing is added by encoding the AID into the message. An IVS which does not support the corresponding application drops the message without further processing. The frequency indicates how often the message shall be distributed to the IVSs in communication range by either an IVS, IRS, or eNodeB. The message is valid until the point in time encoded in expiryTime. After this point in time, all entities will drop and no longer distribute the message. All other fields are applied as in the original GBC/GAC message according to [ETS14a]. The sequence number (*SN*) indicates the index of the sent packet and is utilized to detect duplicate GN packets. The remaining fields are applied to describe the geometric shape of the dissemination area.

This message format can be applied for both LTE and IRS Networks to deliver geocast messages to a group of IVSs. When they are distributed over mobile networks, the GeoNetworking message is the payload of the Internet Protocol (IP) connection. For DSRC, the GeoNetworking message is also used within the Network and Transportation Layer, respectively, to deliver the message. In both cases the receiving IVS parses the GeoNetworking message by its DSRC stack. To check the relevance of a message, the IVS compares the included AID to its list of subscribed AIDs and its current location with the area embedded in the message. Figure 7.4 illustrates the location of the GeoNetworking message in the Open Systems Interconnection (OSI) layers for LTE and DSRC, respectively. Since the creation of this work a new version of the standard on how to secure a message has been published [ETS15]. This version includes the AID of the message in the *Header Fields*. When applying the new version, it is therefore no longer necessary to embed the AID in the extended header.

Due to the presented unified message format, it becomes quite simple for the ICS and IVS to handle messages. The ICS needs to create only one message and can

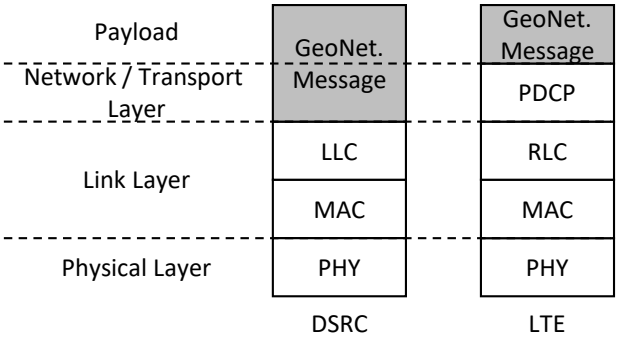


Figure 7.4.: Comparison of the GeoNetworking message location within the LTE and DSRC layers

then forward it to all networks. The IVS can parse the message in the same way, independent of the applied communication technology. Furthermore, an IVS can simply forward a message received by LTE to other IVSs via DSRC without the need to convert it. Therefore, the applied message format enables significant architectural simplifications.

7.3.4. Overhead

When messages are transmitted in a wireless way, all entities in communication range receive the message. Messages are dropped at the network layer when MBMS is applied and the receiver is not part of the corresponding MBMS User Service. In the proposed scheme, each IVS will receive the messages of all ITS applications, neglecting weather it is subscribed to the application or not. This introduces a certain overhead because all received messages need to be forwarded to the DSRC stack. There the messages are dropped if they are not relevant. Whenever messages are received via DSRC, all incoming messages are checked for relevance at network level. This analysis includes an inspection of the messages geographic region. Therefore, this check needs to be extended in order to drop messages from not supported applications at network layer.

Subsequently, there is no overhead introduced when messages are received via DSRC. For LTE each message needs to be forwarded to the DSRC stack. However, these messages are only distributed a few times per minute and have a size of less than 3 Kilobytes. Furthermore, it is not expected that several dozens of messages are valid in the same region at the same time. Therefore, AGfIA clearly does not

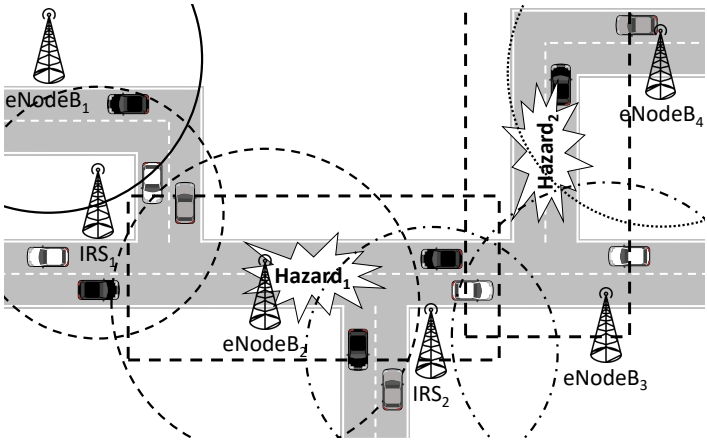


Figure 7.5.: Message distribution example

introduce a significant overhead.

7.3.5. Billing

For economic reasons, it must be possible to bill the network usage of geocast messages. In the outlined scheme, an ICS may pay a basic amount for the service provision by the operators. Furthermore, the ICS might be billed by the number of messages it sends, depending on the size of the dissemination area, sending frequency, and validity period of the messages. Therefore, the network operators do not need any information about the IVS exploiting the messages or even the number of receivers of a message. Accordingly, it is not necessary for the network operators to track the IVSs by exploiting a certain ITS application for billing. Therefore, this scheme protects the IVSs privacy.

7.3.6. Example

We illustrate the advantages of the described geocast scheme using the example in Figure 7.5. The figure shows the two hazards $Hazard_1$ and $Hazard_2$ like an icy road and ongoing roadworks. The rectangles depict the dissemination areas of possible warning messages. Furthermore, the eNodeBs and IRSs covering the area are depicted together with their communication range. When exploiting the proposed scheme, only the eNodeBs and IRSs covering the respective dissemination area by their communication range distribute the message to the IVSs.

In the example IRS_1, IRS_2, $eNodeB_2$, and $eNodeB_3$ cover the dissemination area of $Hazard_1$, while IRS_2, $eNodeB_3$, and $eNodeB_4$ cover the area of $Hazard_2$. Since IRS_2 and $eNodeB_3$ are covering both areas, they distribute both messages. The eNodeBs and IRSs not covering any dissemination area do not forward any geocast message. Therefore, the messages are only disseminated by the relevant eNodeBs and IRSs. Moreover, only the IVSs depicted in black exploit the application warning of $Hazard_1$, while the white ones utilize the application that distributes information about $Hazard_2$. The gray ones illustrate IVSs exploiting both applications. Only the IVSs applying the corresponding application process the warning messages, all other IVSs discard the message. The selection of the relevant IVSs is done without having any knowledge on which IVS exploits any specific ITS application at the network level, nor knowing which IVSs are located within the dissemination area. Therefore, the privacy of each IVS present in the dissemination area is well preserved.

7.4. Implementation

The implementation consists of two Java programs running on the IRS and each IVS, respectively, and was done as part of the work documented in [Bar15b]. Both programs utilize a GeoNetworking stack written in Java. The program running at the IRS gets as input the distribution information from an ICS. As its output it distributes the message via DSRC to the IVSs in communication range.

A screenshot of the GUI running on the IRS is given in Figure 7.6. It displays a list of messages to distribute. It is possible to specify the name, area, AID, repetition interval, expiry time, and payload of a message. Furthermore, it contains a map which shows the targeted dissemination area. For debugging purposes, the GUI also displays a log containing all sent messages. The screenshot shows a basic evaluation scenario which was applied to test if only the messages valid at its current position are considered by the IVS.

As input, the program running on the IVS uses the AID of the running ITS applications, the current position of the IVS, and the received messages. On reception, it parses the messages and checks their relevance. There are three criteria which must be satisfied to consider a message to be relevant: Firstly, the IVS has to be located inside the relevance area. Secondly, it must support the AID. Thirdly, the message at hand must not be expired. The relevance area is encoded into the message and denotes the actual warning region of the event. In general, this area is smaller than the dissemination area.

A screenshot of this program is given in Figure 7.7. It shows the list of AIDs the IVS supports, a list of received messages with their validity state, a log which displays all received messages, and a map containing the IVSs current position and

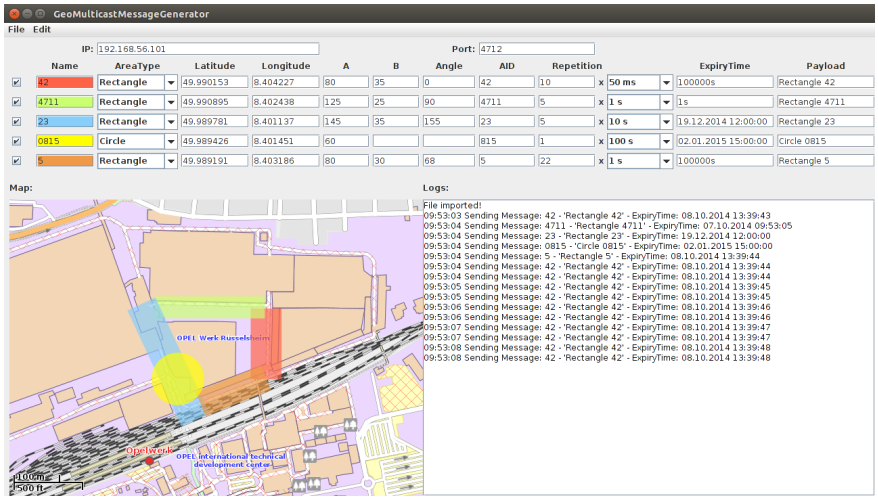


Figure 7.6.: Screenshot of the program running on the IRS

all received messages are displayed. The color of the messages changes with their state. Whenever a message has an AID supported by the IVS and the IVS is located within the relevance area of the message, it is highlighted in green. If the AID is not supported, but the IVS is in its relevance area, it is displayed in yellow. If the receiver is outside of the relevance area of the message, it has a white background. All expired messages are depicted in grey.

7.5. Evaluation

The goal of AGfIA is to distribute messages efficiently in a limited geographic area and in addition preserve the privacy of the participating IVSs. Substantial requirements, as discussed in Section 7.2, are the support for multiple applications, receiver groups, small message transmission, dynamic dissemination areas, validity periods, scalability, a small end-to-end delay, and an efficient transmission.

To check, if those requirements are satisfied, we analyze if they are fulfilled by the proposed scheme. In addition, we compare them, the privacy, system complexity, and supported network types of AGfIA to GBGS [JRX11] and MBMS [3GP13b] as reviewed in Section 3.4. Furthermore, we applied the outlined implementation of AGfIA for IRS Networks to evaluate its properties on top of real-world vehicles.

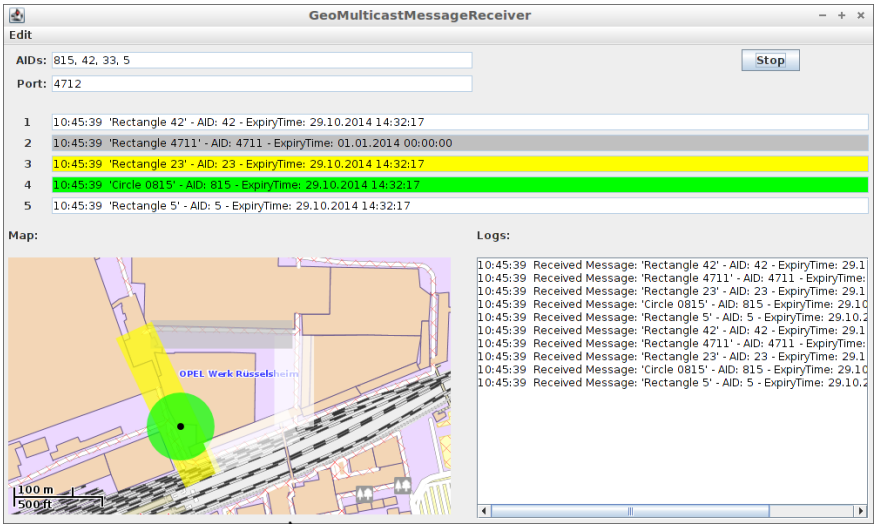


Figure 7.7.: Screenshot of the program running on the IVS

Table 7.1.: Privacy evaluation of the geocast schemes MBMS, GBGS, and AGfIA

	MBMS	GBGS	AGfIA
Position updates	MNO	MNO and GMS	MNO
Utilized applications of IVS	Yes	Yes	No
Subscribed IVSs of application	Yes	Yes	No

7.5.1. Privacy

In order to compare the privacy of the different schemes, we analyze which entities are getting positions updates and if it is possible to get the utilized applications of an IVS or all IVSs subscribed to an application. The results of the evaluation are shown in Table 7.1 and discussed in the sequel.

In each scheme the MNO is able to track the IVSs. To enable a steady connection for clients, the MNO must perform mobility management. This covers packet rerouting and handover management between cells. Therefore, the MNO must know the location of each IVS. However, schemes like Privacy Augmented LTE as proposed in [AKS13] are aimed to prevent tracking by the MNO.

For the grid based geocasting scheme the central GMS is, in addition to the MNO, aware of the positions of each IVS. This is a major privacy drawback compared

Table 7.2.: System Complexity evaluation of the three geocast schemes

	MBMS	GBGS	AGfIA
Registration	Per application	Per application	Once
New area	Additional session	Receiver lookup	eNodeB lookup
Additional network	1 additional message	Nothing	1 additional message
IVS position update	Nothing	Inform central entity	Nothing

to the other two schemes. In case that AGfIA is applied for distribution over IRS Networks, a tracking of the IVSs is technically impossible.

An attacker may aim at identifying all IVSs subscribed to a certain application or at all applications exploited by an IVS. Both GBGS and MBMS maintain a central subscriber database. For MBMS the MNO hast knowledge on which IVS is subscribed to a specific service. When exploiting GBGS the GMS is thus aware of the applications an IVS utilizes. Therefore, an attacker with access to these databases is able to yield the sought information.

In contrast, AGfIA does not maintain such a database. The relationship between the IVSs and applications is not stored anywhere. Therefore, it is not possible to identify the IVSs subscribed to a certain application or to all subscriptions of an IVS. The MNO is only able to determine all IVSs exploiting ITS applications. However, this is not a privacy threat at all since the MNO knows anyway from its contracts which entities are IVSs.

7.5.2. System Complexity

To assess the complexity of the schemes we compared four significant procedures: application registration, geocasting a message in a new area, utilizing an additional network to distribute the messages, and updating the position information of the IVS within the network. The results of the complexity evaluation are shown in Table 7.2.

We analyze the case an IVS wants to register for or unregister from an application. In MBMS and GBGS the IVS must inform a central entity. There, the relation between the IVS and application is either created or deleted. This is not necessary in AGfIA, because each IVS registers itself only once for all ITS applications and not for each application separately. For this use case, AGfIA reduces management overhead and system complexity significantly while preserving the privacy.

For typical ITS applications an ICS must frequently change or add geocast distribution areas. For MBMS, message distribution to new areas requires that a new

session is created. This therefore increases the complexity of the system. When applying GBGS all IVSs in the new area have to be selected at the GMS. This may result in a considerable effort. AGfIA only requires a lookup of all relevant eNodeBs and IRSs in the area. Therefore, AGfIA has a lower complexity than both MBMS and GBGS when messages shall be distributed in a new area.

Messages are distributed via different networks in order to archive a better coverage. When MBMS or AGfIA is applied one additional message needs to be sent for each network. No additional message is necessary when GBGS is applied. There the messages are sent to each IVS individually, independent of the network the IVS is registered at. However, this cannot compensate the overhead introduced by the unicast scheme applied in GBGS and analyzed in the next section.

Due to their high mobility, IVSs frequently change the eNodeB they are connected to. These position updates are handled automatically by the mobile network for all schemes. However, when applying GBGS the IVS has to report its position to the central GMS regularly. Therefore, AGfIA and MBMS have a lower system complexity than GBGS when the position of the IVS changes.

The evaluation shows that AGfIA has a lower complexity than the other schemes. For all four use cases it has the lowest complexity.

7.5.3. Scalability

An ITS geocast message usually addresses a large number of receivers. Therefore, it is important that the applied geocasting scheme does scale with respect to the number of receivers. Hence, we compare the outlined scheme with MBMS and GBGS regarding the number of messages the ICS needs to send to the network operator, the amount of messages within the network of the network operator, the number of messages received by the IRS or eNodeB, and the number of messages sent from the network to the IVSs. The results of the scalability evaluation are shown in Table 7.3, whereas U stands for unicast, M for multicast, and X for Xcast messages, respectively.

For MBMS, the ICS has to pass one message to the network each time a message shall be sent to the IVSs. For the grid based geocasting scheme one message needs to be sent to the network for each IVS at each point in time a hazard message needs to be distributed. Hence, MBMS performs better than GBGS in terms of scalability. When AGfIA is applied, only one message for each hazard needs to be sent to the network no matter of how often it has to be forwarded to the IVSs. Therefore, AGfIA scales better than the other schemes.

For the distribution of the messages within the network, MBMS applies multicast, whereas almost one message is processed by each router. In contrast, one message per receiver is sent in case of GBGS. AGfIA does exploit Xcast to distribute the

Table 7.3.: Scalability evaluation of the three geocast schemes

	MBMS	GBGS	AGfIA
ICS to network	1 per message (Unicast)	1 per message and receiver (Unicast)	1 per hazard (Unicast)
Within network	1 per router (Multicast)	1 per receiver (Unicast)	1 per router (Xcast)
To IRSs / eNodeBs	1 per eNodeB (Multicast)	1 per receiver (Unicast)	1 per eNodeB / IRS (Xcast)
To IVSs	1 per eNodeB (Multicast)	1 per receiver (Unicast)	1 per eNodeB / IRS (Multicast)

messages within the network and needs accordingly almost one message per router. However, the applied Xcast features a larger message header compared to multicast. Therefore, both MBMS and AGfIA scale much better than GBGS within the distributing network.

After the messages are distributed within the network, they are forwarded to the eNodeB or IRS, respectively. In case MBMS or AGfIA is applied, one message is forwarded. When GBGS is applied, one message per receiver is sent to the eNodeBs and IRSs of the network. This results in considerably more messages compared to the other schemes.

If MBMS or AGfIA is applied the messages are distributed by means of multicast from the eNodeBs and IRSs of the network to the IVSs. For the GBGS the messages are distributed by means of unicast to each receiver, which requires more messages compared to multicast.

This evaluation shows clearly that MBMS and AGfIA do scale better than the GBGS with respect to the number of receivers. In these schemes the number of messages does not depend on the number of receivers, but only on the size of the geographic area and on the numbers of eNodeBs and IRSs located within. For AGfIA a larger message header is applied within the network to enable Xcast in comparison to MBMS. However, AGfIA requires only one message per event from the ICS to the MNO. For MBMS the message has to be sent periodically to the MNO, which has to redistribute it in its network in order to reach IVSs entering the area.

7.5.4. Supported Networks

If a scheme is able to support different kinds of networks, it can have a better coverage and therefore it can reach more IVSs in the dissemination area. Furthermore, an ICS might have a better choice of networks to utilize for message distribution.

Table 7.4.: Supported networks of the three geocast schemes

	MBMS	GBGS	AGfIA
Mobile Networks	Yes	Yes	Yes
IRS Networks	No	GN6 only	Yes

Table 7.5.: Comparison of the fulfilled requirements

	MBMS	GBGS	AGfIA
Multiple Applications	o	+	+
Receiver Groups	o	o	+
Content Type	o	+	+
Dissemination Area	-	o	+
Validity Period	o	o	+
Scalability	+	-	+
End-to-End Delay	+	o	+
Efficient Transmission	+	-	+

Our analysis shows that MBMS can be applied to LTE networks only, because there is no support for IRS Networks. GBGS can be utilized for transmissions over LTE and IRS Networks if the IVS and IRS Network do support GN6. In contrast, AGfIA enables the transmission over both LTE and IRS Networks without the limitation on GN6 support. A summary of this comparison is given in Table 7.4

7.5.5. Requirements

We evaluate which of the previously outlined requirements for ITS applications are fulfilled by the different schemes and discuss the results in the sequel. A summary is presented in Table 7.5.

Multiple Applications: All three schemes are able to handle multiple applications. For MBMS there is a bigger effort necessary to support additional applications because a new MBMS user service has to be created first. When employing the other two schemes no such costly operation at the infrastructure side is necessary.

Receiver Groups: Different receiver groups are supported by all these schemes. In case of MBMS and GBGS all interested IVSs must explicitly subscribe in order to be part of the group. For AGfIA no such action is required by an IVS in order to join a group.

Content Type: GBGS and AGfIA are well suited to transmit small ITS messages. MBMS was designed for long-tailed traffic, e. g. multimedia streams. This causes substantial overhead in the ITS scenario featuring short messages. However, a delivery method for short ITS messages may be defined to improve things.

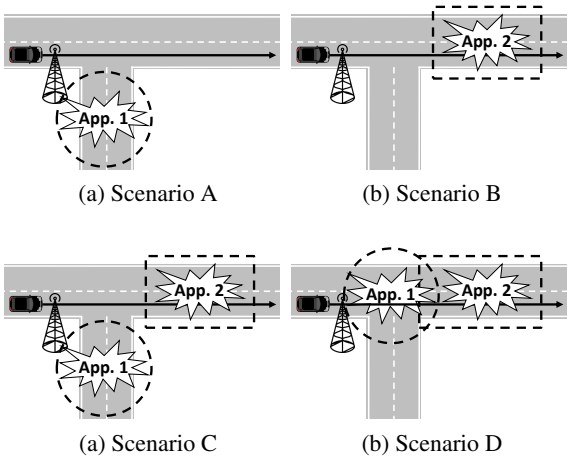
Dissemination Area: For GBGS the dissemination area can be specified for each message. However, the actual distribution area depends on the size of the grids. Accordingly, the area might be much greater than specified. When applying MBMS it is not possible to define the dissemination area in the same granularity. The areas are limited to predefined service areas. Furthermore, it does not support overlapping service areas for the same application. Therefore, messages with overlapping dissemination areas have to be distributed in all service areas covering the dissemination area. In contrast, AGfIA allows to specify the dissemination area for each message separately.

Validity Period: AGfIA repeats the ITS messages in the dissemination area at a given frequency until they expire. Therefore, IVSs arriving in the area also receive older, but still relevant messages. When applying MBMS and GBGS the messages are not repeated in the first place. However, an ICS might either send a message in the desired distribution frequency to the MNO or query the GMS. Then, new arriving IVSs would get the message too. Nonetheless, this introduces unnecessary overhead. To reduce the overhead for the GBGS the ICS can cache the IVSs located in the dissemination area and send the message only to the new IVSs returned by the GMS.

Scalability: As outlined in the previous section, MBMS and AGfIA do scale better than the GBGS.

End-to-End Delay: The end-to-end delay of the distributed messages is smaller for AGfIA and MBMS than for GBGS. In the first two schemes the messages are directly forwarded to the network operators, which distribute the messages to the IVSs. When employing GBGS the geo messaging server introduces an additional delay by identifying the relevant IVSs. Furthermore, one unicast message has to be generated and transmitted for each IVS.

Efficient Transmission: The message transmission scheme applied by GBGS is not very efficient because messages are transported by means of unicast, which introduces more load within the communication networks than Xcast or multicast as applied by MBMS and AGfIA.



(a) Scenario A (b) Scenario B

(a) Scenario C (b) Scenario D

Figure 7.8.: Basic evaluation scenarios

7.5.6. Experimental Evaluation

For the real-world evaluation we utilized the outlined implementation to integrate the proposed scheme into IRS Networks consisting of one IRS and of development vehicles equipped with DSRC. We evaluated the software prototype in several scenarios.

Measurement setup

The evaluation setup consists of two IRS Networks featuring one IRS each. The IRS also runs the IRS CS. Each IVS and IRS consists of an AU and a CCU. The AU runs the application software, whereas the CCU is responsible for sending and receiving DSRC messages. Both units are connected via Ethernet.

Results

Within the outlined setup we evaluated several basic and realistic scenarios. The principles of the basic scenarios are depicted in Figure 7.8. *Scenario A* consists of a message from *Application 1*, which is outside of the IVSs route, whereas in *Scenario B* the message from *Application 2* is on the route of the IVS. *Scenario C* contains messages from different applications, whereas not all are on the envisaged route of the IVS. The reception and processing of multiple overlapping messages from different applications is addressed in *Scenario D*.

We ran several different tests on these basic scenarios to evaluate the prototype implementation. We varied the AIDs an IVS supports from those applications not applied in the scenario up to all AIDs of the messages. To evaluate the message validity check we ran tests with valid messages, expired messages, and messages that will be valid in the future. The evaluation of the different scenarios and configurations showed that the IVSs running the particular application consider a message as relevant only in case that they are within the relevance area of the message and the message is still valid.

As an example for a realistic and complex evaluation, a scenario around Rüsselsheim, Germany is depicted in Figure 7.9. This scenario features two IRS Networks, *IRS*1 and *IRS*2, respectively. A possible route of an IVS is drawn in red aiming from Rüsselsheim city towards a motorway.

In this scenario three messages denoted as $message_1$, $message_2$, and $message_3$ are distributed to the IVSs, each with a different AID. The relevance area of the messages is indicated by a filled shape surrounding the message name. The larger frame indicates the dissemination area of the messages. $message_1$ exemplifies an icy road an icy road warning. Therefore, its shape is enclosing the icy. The dissemination area has the same shape but covers a much larger area in order to inform approaching IVSs before they reach the hazard. Only IRS_1 is located within the dissemination area. Therefore, only this IRS is distributing the message. $message_2$ located in the center of the city represents a notification about road closures due to a public street festival. To inform all IVSs reaching the center, the dissemination area covers the whole city. Because of the large dissemination area, $message_2$ is in reach of both IRSs and is therefore distributed by all of them. $message_3$ warns about a traffic jam eastbound on the highway located south to the city. Therefore, the dissemination area is only extended towards west, where the IVSs reach the traffic jam. Only IRS_2 is located within the dissemination area of this message and distributes it.

This scenario clearly demonstrates one of the main advantages of AGfIA: Only the IRSs and eNodeBs located within the dissemination area of a certain message distribute this message. In contrast, even IRSs or eNodeBs in overlapping areas distribute all relevant messages. In addition, only the IVSs exploiting the corresponding application process the message. All other IVSs drop the messages at network layer. To sum up, we demonstrated that AGfIA works well on both real devices and in a complex traffic scenario.

7.6. Summary

In this chapter we proposed AGfIA, an AGfIA. This scheme exploits various communication technologies in order to send messages from an ICS to all IVSs which

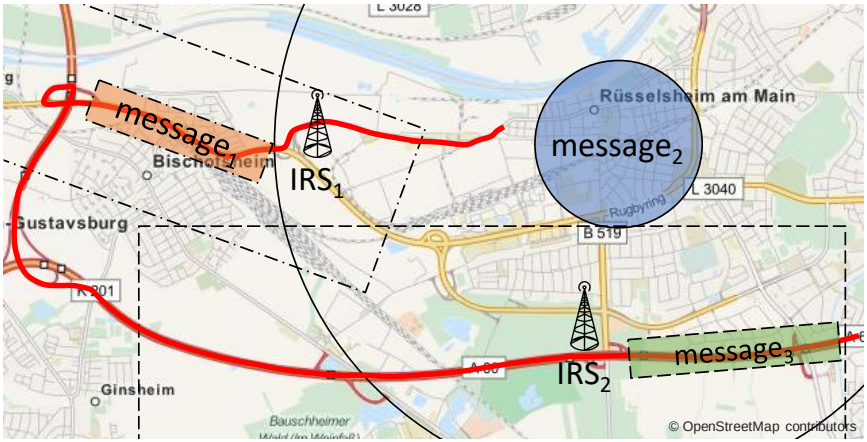


Figure 7.9.: Example of a real-world evaluation scenario

belong to a distinct group, like subscribed to a specific application, and are located in a certain area. Hereby, the message receivers stay unknown for privacy reasons. Moreover, no entity is able to determine the applications exploited by a certain IVS or to identify all IVSs subscribed to a specific ITS application. Furthermore, the same message format can be applied for rather different communication technologies. We detailed on how the scheme works in terms of registration, message distribution, billing, and how the proposed message format looks like.

As message format we took GeoNetworking as standardized for DSRC and extended it to support multicast aimed to efficiently address a group of IVSs. In comparison to other geocast schemes, the presented AGfIA approach protects the privacy of the IVSs. An attacker is not able to discover the applications exploited by an IVS or to identify all IVS subscribed to a certain application. Furthermore, this scheme does not rely on a central entity, which may track the location of the IVSs. Therefore, the scheme protects the privacy of association and location. By not tracking the IVSs nor storing their identity somewhere the privacy of the identity is also protected. Furthermore, AGfIA reduces the system complexity compared to other known geocast schemes. By exploiting AGfIA it is simple to register for ITS applications, to distribute a message in a new area, or to disseminate the message via a new network. Furthermore, no additional effort is necessary, if the location of an IVS changes. AGfIA scales in addition for a large number of receivers or for huge dissemination areas. Moreover, messages can be distributed by means of different communication technologies like LTE or DSRC. Its major advantage is that it fulfills substantial requirements of ITS geocast applications. To secure the communication the ICS can

authenticate the messages with its certificate. Furthermore, it might distribute an encryption key to all subscribed IVSs in order to ensure the confidentiality of the transmitted data.

Additionally, we created a prototype implementation of the scheme for DSRC, set up experimental IRS Networks, and evaluated the prototype software on top of real IVSs. The presented results show that the AGfIA system is well-suited for an efficient distribution of ITS messages to a group of IVSs located within a certain area. At the same time AGfIA protects the privacy of the IVS.

In comparison to existing geocast schemes the one proposed in this chapter is well suited for ITS applications. Its novelty is that it is optimized for small messages, scales for a large number of receivers, features a low end-to-end delay, supports dynamic dissemination areas, and protects at the same time the privacy of the IVSs. Existing schemes do not fulfill all these important properties for ITS applications. Most properties of the proposed scheme are achieved by caching the messages at the edge of the network and applying a single multicast group for all IVSs.

The proposed anonymous geocast scheme might be exploited in two different scenarios. Either an ICS is distributing information to the IVSs or it is requesting data from the IVSs. Data distributed to the IVSs might be for example map data updates for automated driving or warnings about weather hazards. Requests for sensor data in a certain region might be applied to, e.g., validate weather information or changed street conditions. Moreover, it might be extended to support further communication technologies. The outlined scheme allows message dissemination via DSRC and LTE. However, it might be extended to support other technologies like consumer WLAN.

8 | Conclusions

8.1. Summary

Connected vehicles and automated driving are two current trends in the automotive industry which will change the future driver experience. Connected vehicles exchange data for e.g. convenience or entertainment applications with central servers or other vehicles. Automated driving vehicles require up-to-date high-precision map data in order to fully utilize all benefits. Therefore, this data needs to be also exchanged with central servers. All this data, from connected and automated driving vehicles might reveal sensible private information about the driver.

We investigated in this thesis four different vehicular communication scenarios. In all of these scenarios challenges resulting from different requirements originating from vehicle and application operators needed to be solved. The application requires in general reliable, accurate, and fresh data while the operator of the vehicles aims in preserving its privacy. For each scenario we proposed a solution to solve these challenges. The investigated scenarios include the authentication of a vehicle at a central station, the upload of sensor data from a vehicle to a central station, the secure privacy preserving communication between vehicles, and the distribution of information form a central server to vehicles located in a geographic region.

The first scheme allows a vehicle to anonymously authenticate at a central server based on properties. When an operator creates a new application, he has to specify the properties the vehicles need to possess in order to use the application. To exploit the application, the vehicles first needs to obtain a ticket from a central entity. This ticket can then be exploited to authenticate for the application. To get this ticket, the vehicles proofs without revealing any other properties to the central entity that it possesses the necessary properties. Therefore, the central stations do learn only the necessary properties. Furthermore, it is not possible to determine all applications exploited by a certain vehicle or to link a ticket to a specific vehicle. Subsequently, this scheme protects the privacy of the vehicles. Additionally, we detailed how billing can be integrated into this scheme and the communication between the different en-

tities can be secured. Furthermore, we showed by a prototype that the scheme does not introduce any noteworthy delay, even with limited computational power on the vehicle side.

The second contribution is a protocol to establish a secured communication channel between two vehicles. The protocol enables two vehicles eligible for the same application to authenticate and agree on an encryption key without leaking their identity. This key can then be exploited to exchange confidential data. To proof the eligibility, the vehicles authenticate each other with tickets obtained by the first proposed scheme. These tickets neither leak the identity, attributes, nor all associations of the vehicle. In order to preserve the communication privacy of the vehicles and establish a secure connection, we rely on a special signature scheme which hides the creator of a signature and a mechanism to derive an encryption key from public available information, respectively. Accordingly, this protocol enables two vehicles running the same application to exchange confidential application specific data without exposing their identity to anyone. We showed by simulation that the applied signatures enhance the privacy of the vehicles. An attacker is not able to link different executions of the protocol to the same vehicle, even when the exploited tickets are reused. Furthermore, we implemented the protocol on real vehicles and exchanged real sensor data. The evaluation of the implementation also shows that the protocol can be executed within a small time frame. Therefore, we also demonstrated that the protocol works well on real devices. Thus, the proposed scheme enables two vehicles to secure exchange data while preserving their privacy.

The third contribution focuses on preserving the privacy of vehicles which send enhanced sensor data to a central server. Sensor data collected by vehicles is usually only valid in a small area and short period of time. Therefore, the data sent from a vehicle to a central server usually contains this information. However, this information might be exploited to harm the privacy of the vehicle. The proposed scheme protects the privacy of the vehicle by exchanging the sensor data with other vehicles prior to sending it to the central server. For the exchange, we apply our proposed protocol which preserves the privacy of the participating vehicles. By exchanging the sensor data, an attacker at the central server is no longer able to identify the vehicles. We proposed different strategies on how to exchange the data between vehicles. Furthermore, we showed by simulation that this scheme hides the identity of the vehicle collecting the sensor reading. The results show that the scheme obfuscates the spatial information embedded into the reported data. In addition, it also obscures the identity of the reporting vehicle. At the same time, this scheme introduces only a short temporal delay. Furthermore, all reported data is authenticated. Therefore, this scheme is well suited to hide the origin of sensor readings when they are sent from a vehicle to a central server. Subsequently, it protects the vehicles privacy.

Fourthly, we developed a scheme to send a message to all vehicles subscribed to a

specific application and located in a certain geographical area. This message distribution is done without knowing the vehicles present in the target area. Furthermore, in contrast to the state of the art solutions, it is not possible to get information of the used application of a vehicle. Therefore, this scheme protects the privacy of the vehicle. In addition, it scales better and has a lower system complexity than state of the art solutions. Moreover, it supports secure communication via different technologies. We created a proof of concept implementation for one communication technology in order to show that the scheme works in real world scenarios. Hence, the proposed scheme is well suited to send a message to all vehicles located in a specific area and exploiting a certain application while preserving their privacy.

All four schemes developed in this thesis have been implemented and tested on real hardware like development vehicles. Therefore, we demonstrated that all proposed schemes also work on real world devices. In addition, we run simulations for some of the schemes in order to evaluate the privacy aspects. We showed for each presented scheme that it protects the privacy of the vehicle on the considered communication scenario. At the same time, they do not introduce a disadvantage in its usability. They also do not violate any important demands of the vehicle or central server. Furthermore, all schemes are complaint with the current standards for vehicular communication. Therefore, it is easy to exploit the scheme in production vehicles.

The proposed schemes might not only be applied separately. Moreover, it is possible to employ several or even all of them at the same time in order to protect the different communication scenarios of an application. Some of the schemes even require the others in order to exploit all their features.

8.2. Future Work

The communication scenarios considered in this thesis are only a subset of all. There exist other scenarios which might introduce additional security challenges to the communication system. One of these scenarios is the request of data in a certain region. From the spatial information in the request a central server might be able to determine the coarse location of the vehicle and therefore track its movements.

Currently there are no practical mechanisms to detect misbehaving vehicles. This includes the collection of the information necessary to make a decision, ensuring the made decision is valid, and to prevent further communication of the vehicle.

The security of the applied mechanisms might decrease over time when new attacks are developed or attackers have more computational power. Therefore, processes to introduce new security mechanisms in vehicles which might be more than ten years old and have very limited hardware capabilities need to be developed. It is

unlikely that all vehicles can be updated at the same time, therefore mechanisms are necessary which ensure the interoperability between older and newer vehicles.

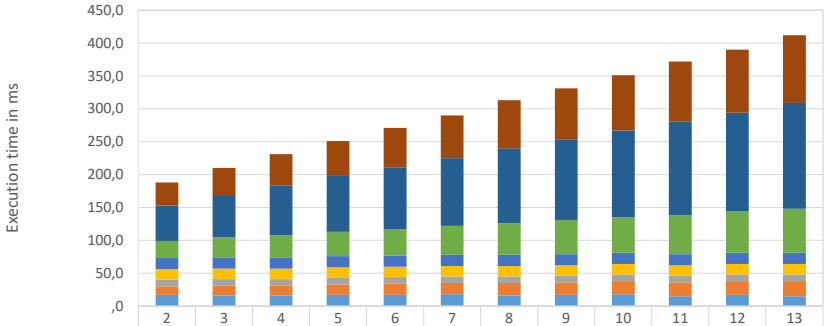
All these issues need to be addressed in future research in order to make the whole system more sustainable.

A | Additional Evaluation Results

A.1. Anonymous Data Exchange

Figure A.1 compares the different protocol steps, transmission, and other operations for ring sizes from 2 to 13. The higher execution times for larger ring sizes are mainly caused by the additional time necessary to create and validate the ring signatures.

The Figures A.2 till A.12 illustrate the time necessary to execute the individual parts of the protocol steps for ring sizes from 2 to 12.



	2	3	4	5	6	7	8	9	10	11	12	13
Step 1	35	41	48	53	60	65	73	78	84	91	96	103
Step 2	54	64	75	85	94	103	114	122	132	143	150	161
Step 3	26	31	34	37	40	44	48	52	54	59	63	67
Step 4	17	17	17	17	17	17	17	17	17	17	17	17
Step 5	16	16	16	16	16	16	16	16	16	16	16	16
Validation Step 5	10	10	10	10	10	10	10	10	10	10	10	10
Transmission	13	15	15	16	17	17	19	19	20	21	21	23
Other	17	16	16	17	17	18	16	17	18	15	17	15

Figure A.1.: Comparison of the protocol steps for different ring sizes.

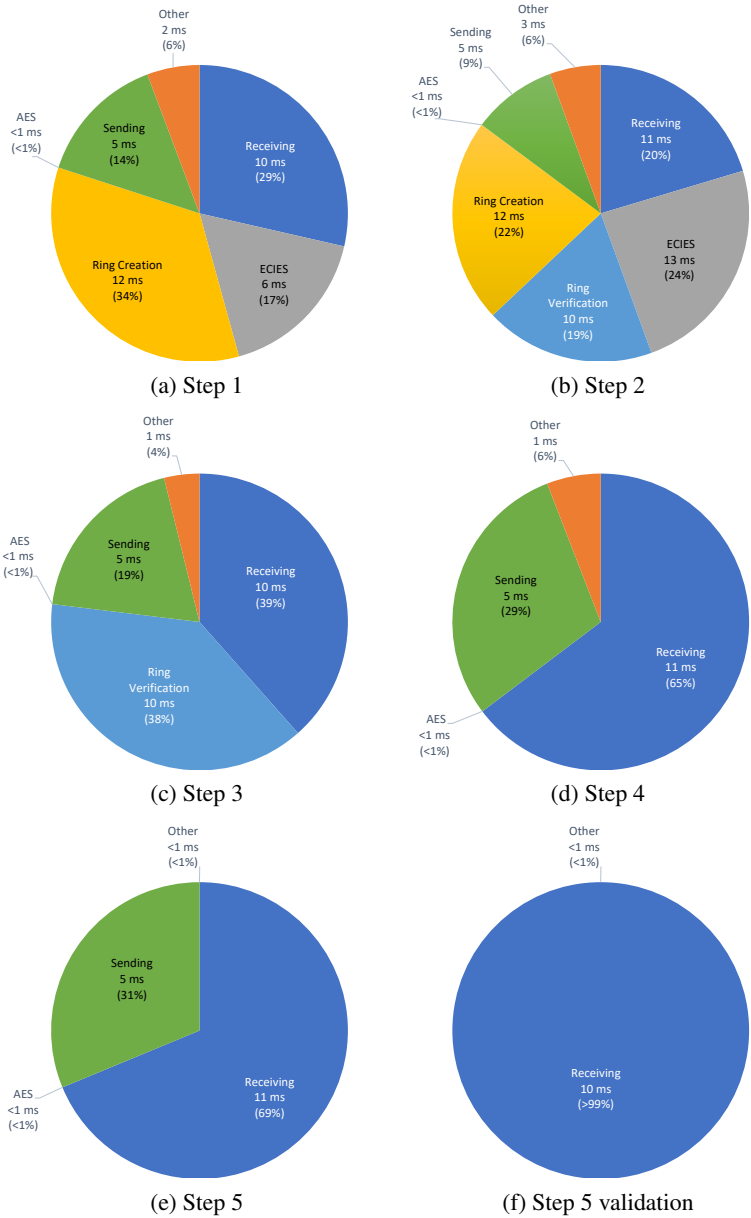
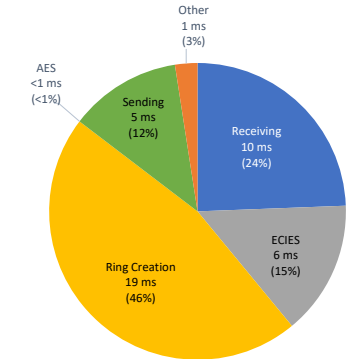
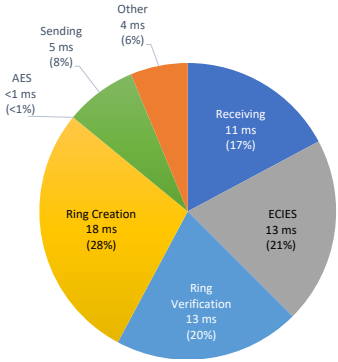


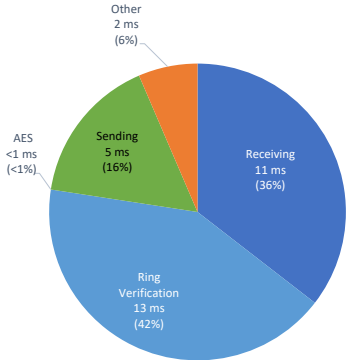
Figure A.2.: Necessary time to execute the individual parts of the protocol steps for a ring size of 2.



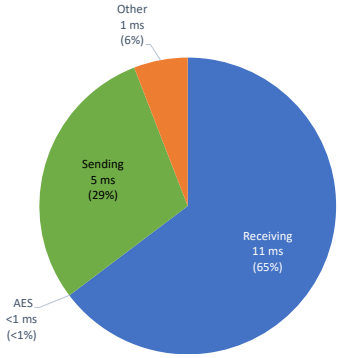
(a) Step 1



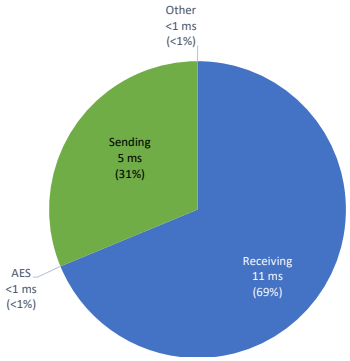
(b) Step 2



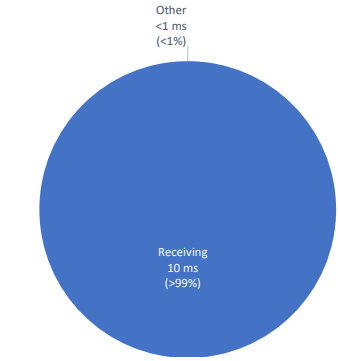
(c) Step 3



(d) Step 4

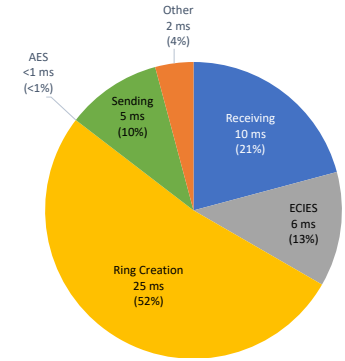


(e) Step 5

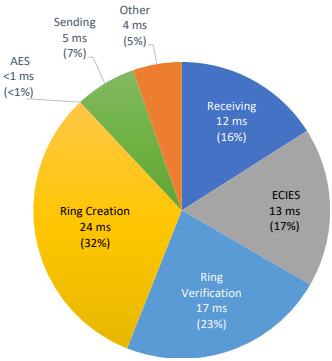


(f) Step 5 validation

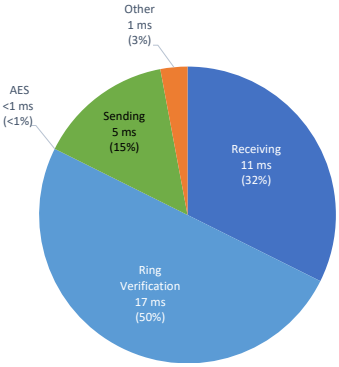
Figure A.3.: Necessary time to execute the individual parts of the protocol steps for a ring size of 3.



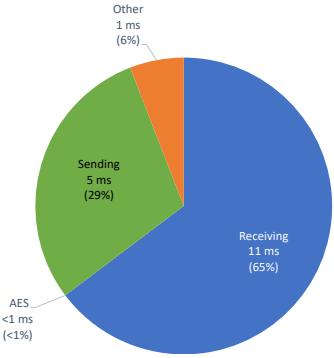
(a) Step 1



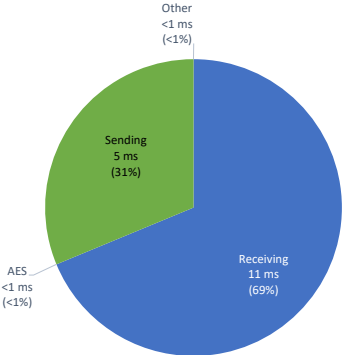
(b) Step 2



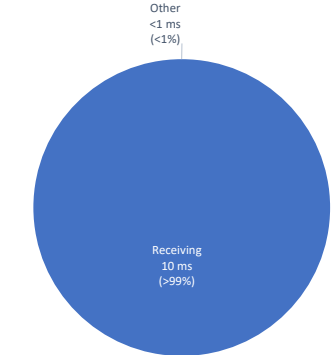
(c) Step 3



(d) Step 4



(e) Step 5



(f) Step 5 validation

Figure A.4.: Necessary time to execute the individual parts of the protocol steps for a ring size of 4.

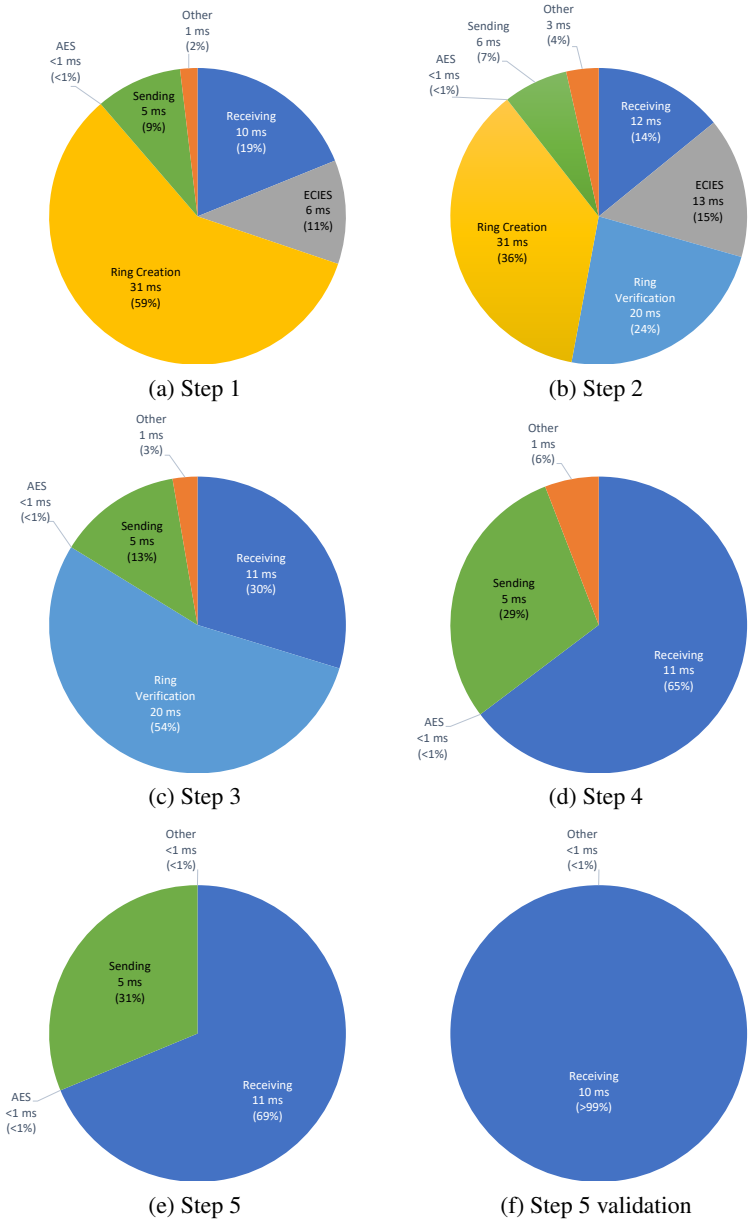
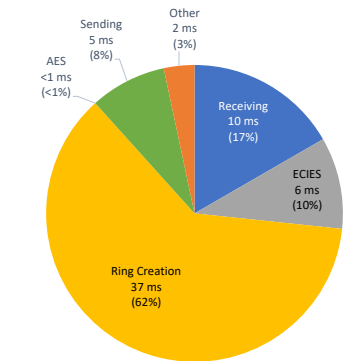
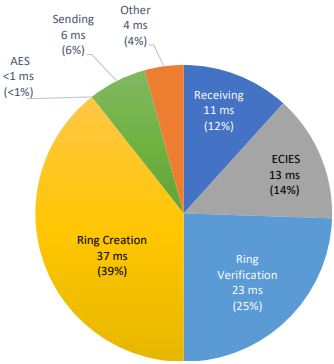


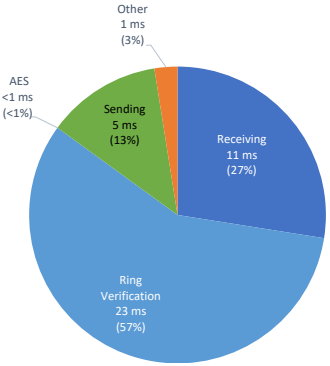
Figure A.5.: Necessary time to execute the individual parts of the protocol steps for a ring size of 5.



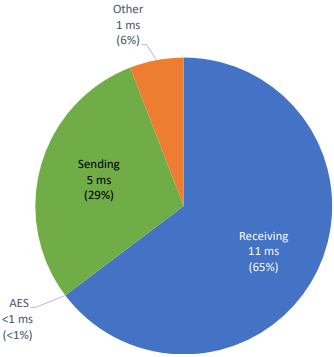
(a) Step 1



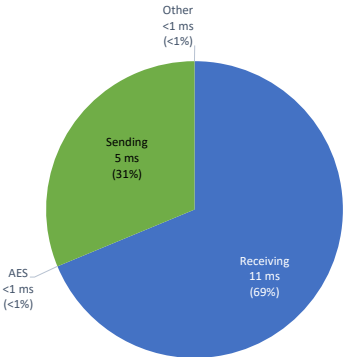
(b) Step 2



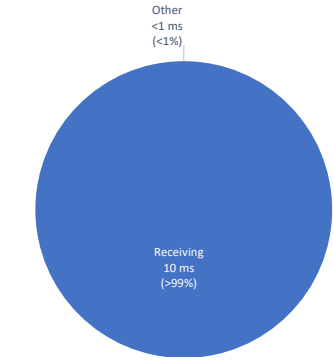
(c) Step 3



(d) Step 4

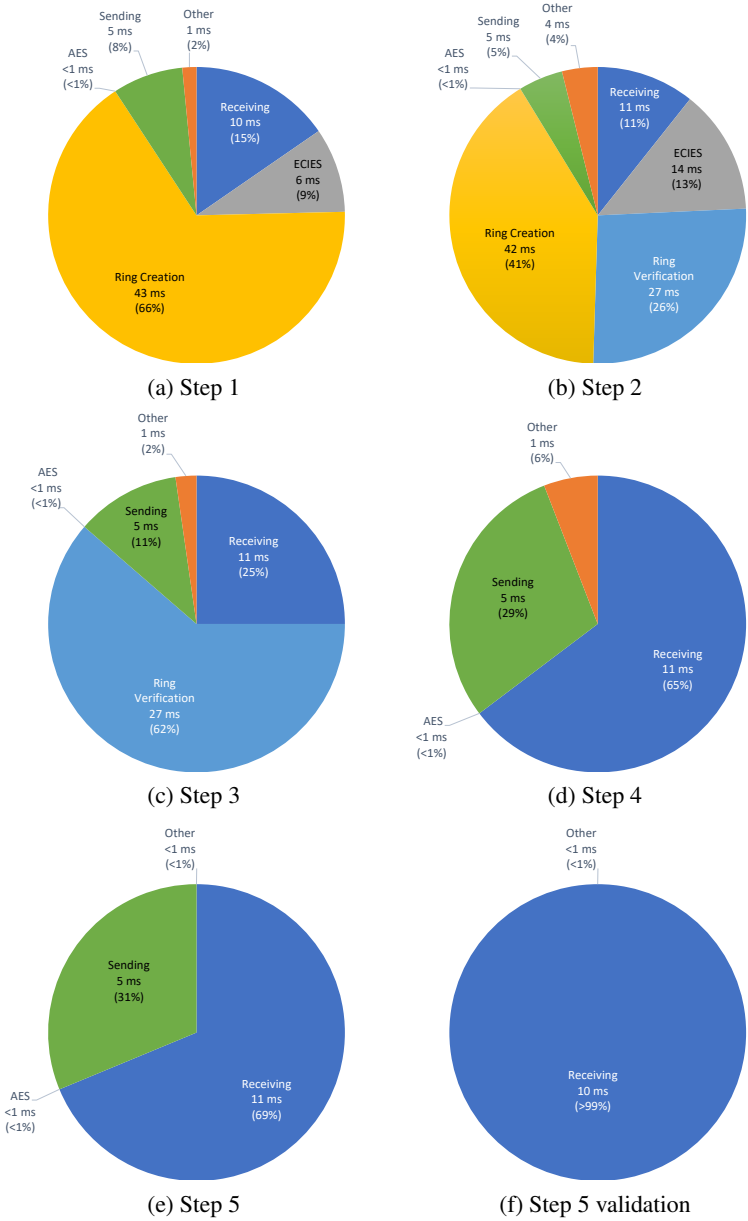


(e) Step 5



(f) Step 5 validation

Figure A.6.: Necessary time to execute the individual parts of the protocol steps for a ring size of 6.



(a) Step 1

(b) Step 2

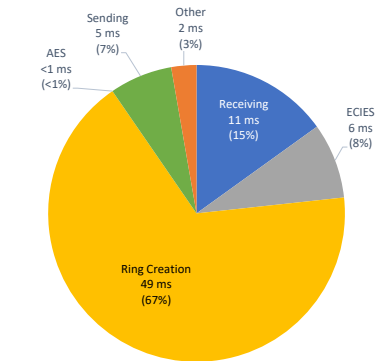
(c) Step 3

(d) Step 4

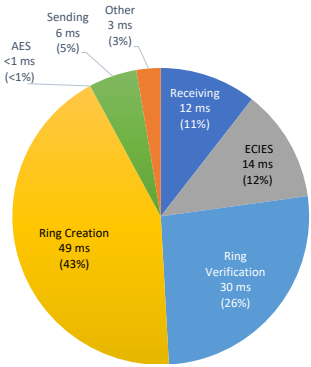
(e) Step 5

(f) Step 5 validation

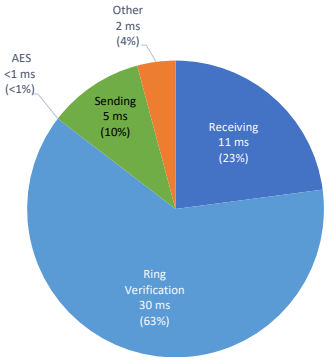
Figure A.7.: Necessary time to execute the individual parts of the protocol steps for a ring size of 7.



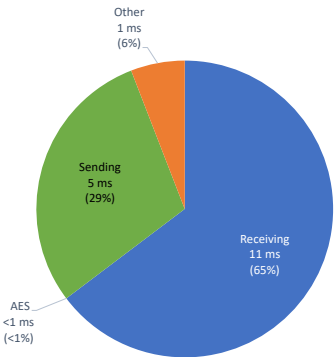
(a) Step 1



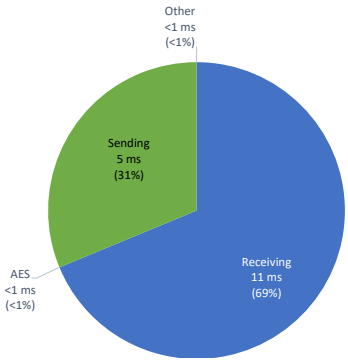
(b) Step 2



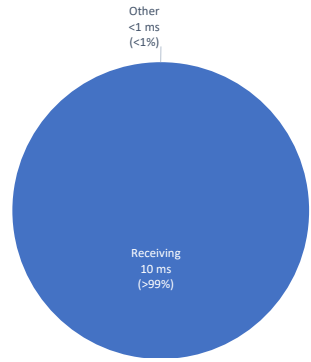
(c) Step 3



(d) Step 4

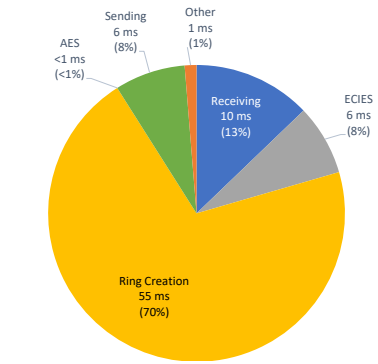


(e) Step 5

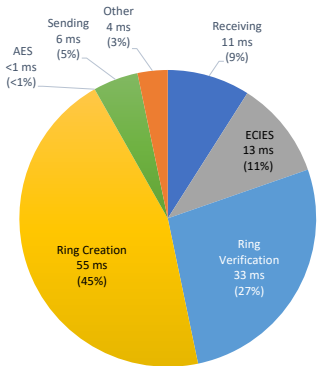


(f) Step 5 validation

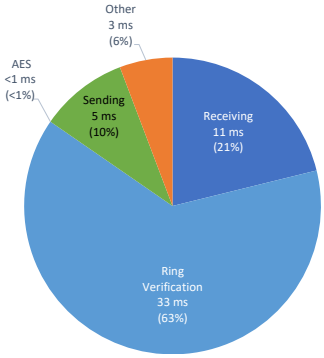
Figure A.8.: Necessary time to execute the individual parts of the protocol steps for a ring size of 8.



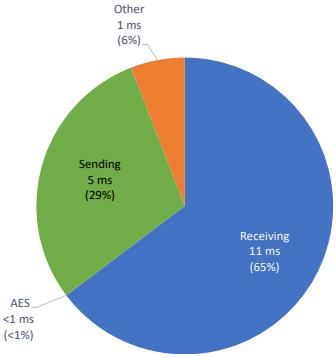
(a) Step 1



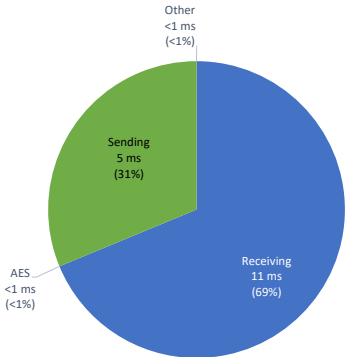
(b) Step 2



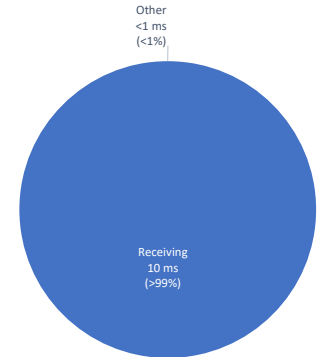
(c) Step 3



(d) Step 4

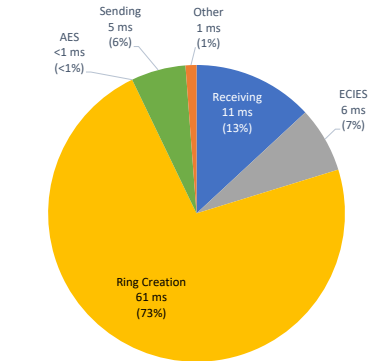


(e) Step 5

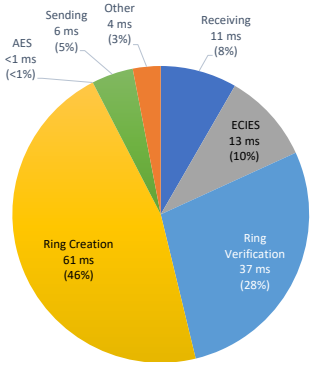


(f) Step 5 validation

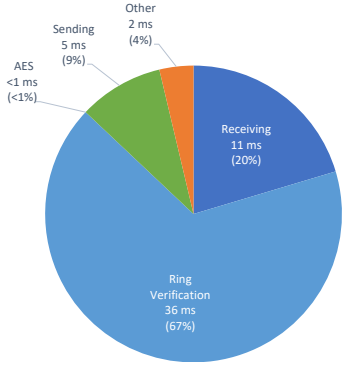
Figure A.9.: Necessary time to execute the individual parts of the protocol steps for a ring size of 9.



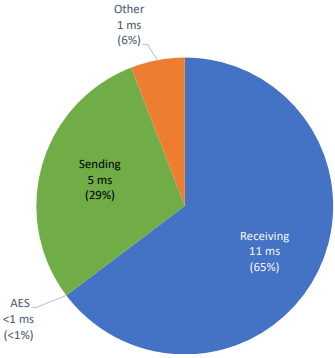
(a) Step 1



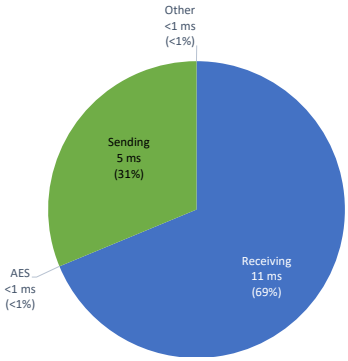
(b) Step 2



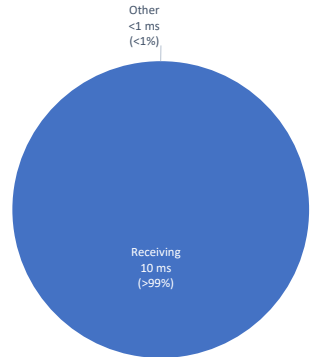
(c) Step 3



(d) Step 4

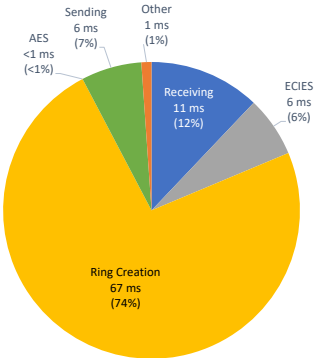


(e) Step 5

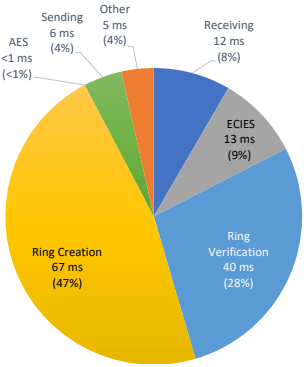


(f) Step 5 validation

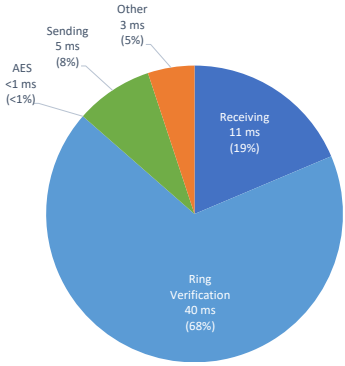
Figure A.10.: Necessary time to execute the individual parts of the protocol steps for a ring size of 10.



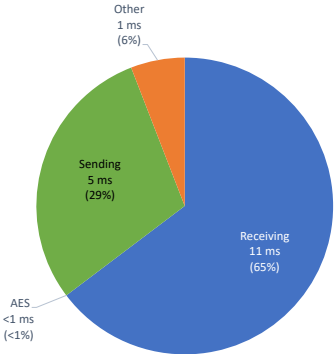
(a) Step 1



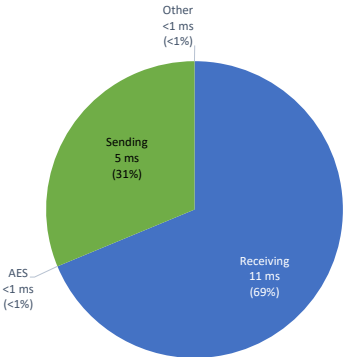
(b) Step 2



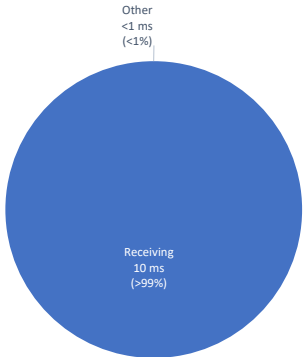
(c) Step 3



(d) Step 4

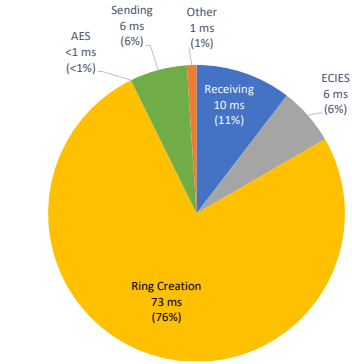


(e) Step 5

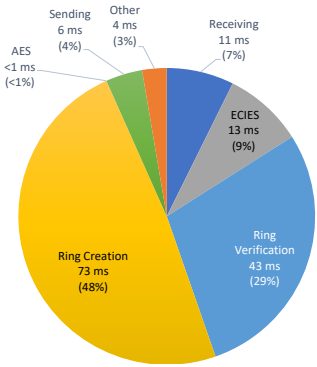


(f) Step 5 validation

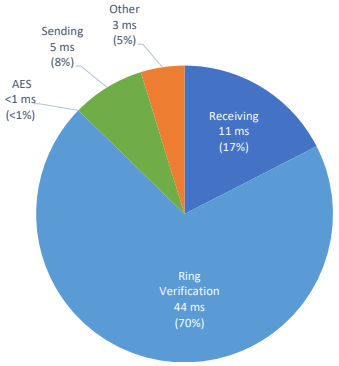
Figure A.11.: Necessary time to execute the individual parts of the protocol steps for a ring size of 11.



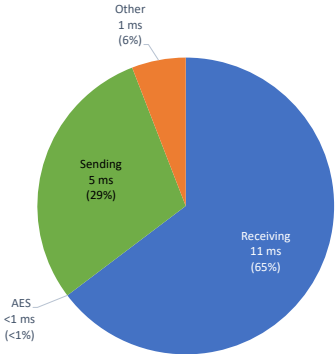
(a) Step 1



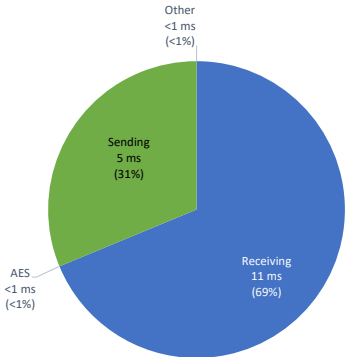
(b) Step 2



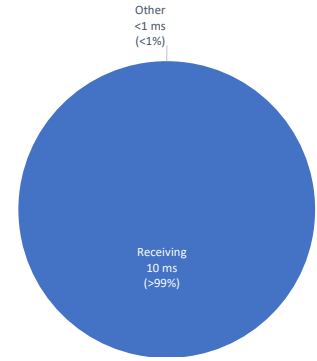
(c) Step 3



(d) Step 4



(e) Step 5



(f) Step 5 validation

Figure A.12.: Necessary time to execute the individual parts of the protocol steps for a ring size of 12.

A.2. Anonymous Data Reporting

Figure A.13 shows the resulting k-anonymity, spatial obfuscation, upload delay, and exchanges per vehicle in one hour of the simulation for a *DataDuplication* value of 1 and Figure A.14 for a value of 5.

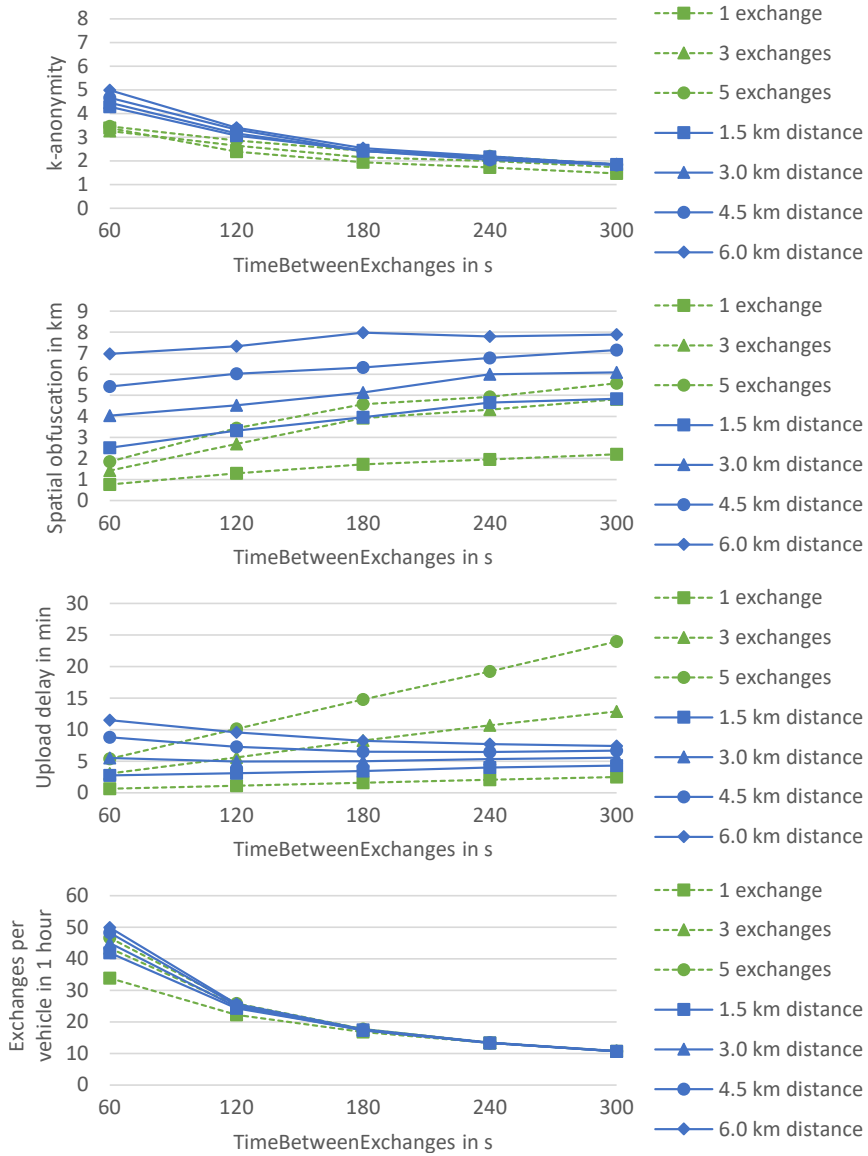


Figure A.13.: k-anonymity, spatial obfuscation, upload delay, and the number of exchanges per vehicle in one hour as a function of *TimeBetweenExchanges* for a *DataDuplication* value of 1.

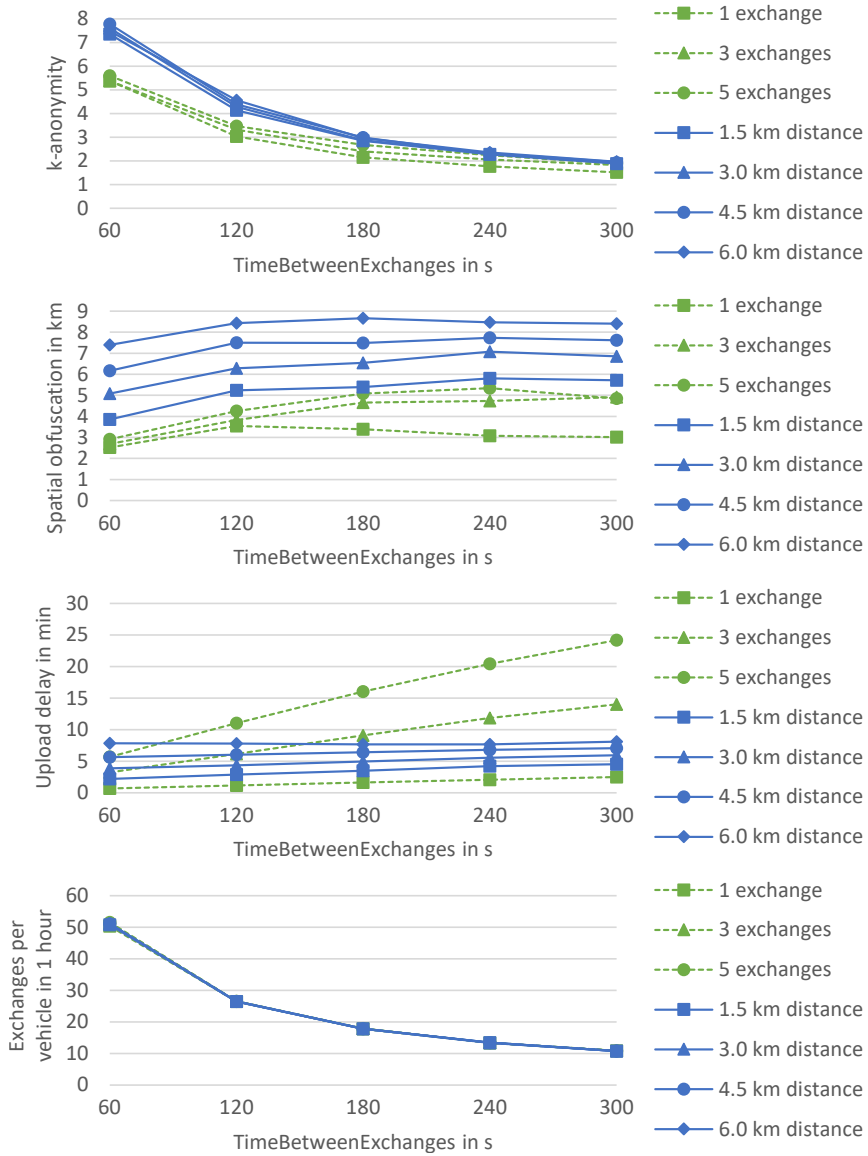


Figure A.14.: k-anonymity, spatial obfuscation, upload delay, and the number of exchanges per vehicle in one hour as a function of *TimeBetweenExchanges* for a *DataDuplication* value of 5.

Bibliography

[3GP13a] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2 (Release 12). 3GPP TS 36.300 V12.0.0, 3rd Generation Partnership Project, December 2013.

[3GP13b] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Multimedia Broadcast/Multicast Service (MBMS); Architecture and functional description (Release 12). 3GPP TS 23.246 V12.0.0, 3rd Generation Partnership Project, December 2013.

[ABC] ABC4Trust. Project Website. `https://www.abc4trust.eu` (accessed in 04.2016).

[ACC+13] G. Araniti, C. Campolo, M. Condoluci, A. Iera, and A. Molinaro. LTE for vehicular networking: a survey. *IEEE Communications Magazine*, 51(5):148–157, 2013.

[AHO08] E. Aimeur, H. Hage, and F.S.M. Onana. Anonymous Credentials for Privacy-Preserving E-learning. In *Proc. of the 2008 International MCETECH Conference on e-Technologies*, pages 70–80, 2008.

[AKS13] D. Angermeier, A. Kiening, and F. Stumpf. PAL - Privacy Augmented LTE: A Privacy-preserving Scheme for Vehicular LTE Communication. In *Proc. of the 10th ACM International Workshop on Vehicular Inter-networking, Systems, and Applications (VANET)*, pages 1–10, 2013.

[APPR09] I. Armac, A. Panchenko, M. Pettau, and D. Retkowitz. Privacy-Friendly Smart Environments. In *Proc. of the third International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST)*, pages 425–431, 2009.

[AST10] ASTM. Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle Systems — 5 GHz Band Dedicated Short Range Communications (DSRC) Medium Access Control (MAC) and Physical Layer (PHY) Specifications. ASTM E2213-03, ASTM International, 2010.

[Azi15] Y. Aziza. Anonymous Credentials in Car-to-X Communication. Master's thesis, Technische Universität Darmstadt, 2015.

[Bar04] R. Barr. SWANS - Scalable Wireless Ad hoc Network Simulator User Guide. `http://jist.ece.cornell.edu/docs/040319-swans-user.pdf` (accessed in 04.2016), 2004.

[Bar15a] F. Bartels. Implementation and Evaluation of an Anonymous Authenticated Key Agreement Protocol in Vehicular Ad Hoc Networks. Bachelor's Thesis, University of Applied Sciences Wiesbaden Rüsselsheim, 2015.

[Bar15b] F. Bartels. Senden und Empfangen von C2X-GeoMulticast-Nachrichten. Technical report, University of Applied Sciences Wiesbaden Rüsselsheim, 2015.

[BBH15] C. Büttner, F. Bartels, and S. A. Huss. Real-World Evaluation of an Anonymous Authenticated Key Agreement Protocol for Vehicular Ad-Hoc Networks. In *Proc. of the 11th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 651–658, 2015.

[BCC04] E. Brickell, J. Camenisch, and L. Chen. Direct Anonymous Attestation. In *Proc. of the 11th ACM Conference on Computer and Communications Security (CCS)*, pages 132–145, 2004.

[BDS+03] D. Balfanz, G. Durfee, N. Shankar, D. Smetters, J. Staddon, and Hao-Chi Wong. Secret handshakes from pairing-based key agreements. In *Proc. of the Symposium on Security and Privacy*, pages 180–196, 2003.

[BFI+07] R. Boivie, N. Feldman, Y. Imai, W. Livens, and D. Ooms. Explicit Multicast (Xcast) Concepts and Options. RFC 5058, 2007.

[BG85] R. W. Baldwin and W. C. Gramlich. Cryptographic Protocol for Trustable Match Making. In *Proc. of the 1985 IEEE Symposium on Security and Privacy*, pages 92–92, 1985.

[BH14] C. Büttner and S. A. Huss. Anonymous credentials and attribute-based authorization tickets in car-to-x communication. In *Proc. of the 2nd GI/ITG KuVS Fachgespräch Inter-Vehicle Communication (FG-IVC 2014)*, pages 9–12, 2014.

[BH15a] C. Büttner and S. A. Huss. Path Hiding for Privacy Enhancement in Vehicular Ad-Hoc Networks. In *Proc. of the 82nd IEEE Vehicular Technology Conference (VTC Fall)*, pages 1–5, 2015.

[BH15b] C. Büttner and S. A. Huss. Verfahren zum Verbreiten einer Nachricht. Patent application, DE 102015009599.4, 2015.

[BH15c] C. Büttner and S. A. Huss. A Novel Anonymous Authenticated Key Agreement Protocol for Vehicular Ad Hoc Networks. In *Proc. of the 1st International Conference on Information Systems Security and Privacy (ICISSP)*, pages 259–269, 2015.

[BH15d] C. Büttner and S. A. Huss. *Information Systems Security and Privacy: First International Conference, ICISSP 2015, Angers, France, February 9-11, 2015, Revised*

Selected Papers, chapter An Efficient Anonymous Authenticated Key Agreement Protocol for Vehicular Ad-Hoc Networks Based on Ring Signatures and the Elliptic Curve Integrated Encryption Scheme, pages 139–159. 2015.

[BH16a] C. Büttner and S. A. Huss. An Anonymous Geocast Scheme for ITS Applications. In *Proc. of the 2nd International Conference on Information Systems Security and Privacy (ICISSP)*, 2016.

[BH16b] C. Büttner and S. A. Huss. Attribute-Based Authorization Tickets for Car-to-X Communication. In *Proc. of the 2015 IEEE Conference on Communications and Network Security (CNS)*, 2016.

[BH16c] C. Büttner and S. A. Huss. *Information Systems Security and Privacy: Second International Conference, ICISSP 2016, Rome, Italy, February 19-21, 2016, Revised Selected Papers*, chapter An Efficient Approach to Anonymous Distribution of ITS Messages in Geographic Areas via LTE and IRS Networks. 2016. to appear.

[BL10] E. Brickell and J. Li. Enhanced Privacy ID from Bilinear Pairing for Hardware Authentication and Attestation. In *Proc. of the 2nd IEEE International Conference on Social Computing (SocialCom)*, pages 768–775, 2010.

[Blo14] Bloomberg. Global market share of the world's largest automobile OEMs as of August 30, 2014. `http://www.statista.com/statistics/316786/global-market-share-of-the-leading-automakers/` (accessed in 04.2016), 2014.

[BMW] BMW. ConnectedDrive Services. `http://www.bmw.de/de/topics/faszination-bmw/connecteddrive/services-apps/connecteddrive-services.html` (accessed in 04.2016).

[Bre14] C. Brenner. Sensordatenübertragung. Bachelor's Thesis, University of Applied Sciences Wiesbaden Rüsselsheim, 2014.

[BSS+11] N. Bißmeyer, H. Stuebing, E. Schoch, S. Götz, J. P. Stotz, and B. Lonc. A generic Public Key Infrastructure for securing Car-to-X Communication. In *Proc. of the 18th ITS World Congress*, 2011.

[Buc04] J. Buchmann. *Introduction to Cryptography*. 2004.

[CEN04] CEN. Road transport and traffic telematics - Dedicated short-range communication - Physical layer using microwave at 5,8 GHz. CEN EN 12253, European Committee for Standardization, 2004.

[CEN15a] CEN/ISO. Intelligent transport systems - Cooperative ITS - Dictionary of in-vehicle information (IVI) data structures. CEN/ISO TS 19321, International Organization for Standardization / European Committee for Standardization, 2015.

[CEN15b] CEN/ISO. Intelligent Transport Systems - Cooperative ITS - Using V2I and I2V Communications for Applications Related to Signalized Intersections (SPaT, MAP, SRM, SSM). Draft CEN/ISO TS 19091, International Organization for Standardization / European Committee for Standardization, 2015.

[CGR+11] D. Christin, J. Guillemet, A. Reinhardt, M. Hollick, and S.S. Kanhere. Privacy-Preserving Collaborative Path Hiding for Participatory Sensing Applications. In *Proc. of the 8th IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, pages 341–350, 2011.

[CH91] D. Chaum and E. Heyst. *Advances in Cryptology - EUROCRYPT 91: Proc. of the Workshop on the Theory and Application of Cryptographic Techniques*, volume 547, chapter Group Signatures, pages 257–265. 1991.

[Cha82] D. Chaum. Blind Signatures for Untraceable Payments. In *Advances in Cryptology: Proc. of Crypto 82*, pages 199–203, 1982.

[Cha85] D. Chaum. Security Without Identification: Transaction Systems to Make Big Brother Obsolete. *Communications of the ACM*, 28(10):1030–1044, 1985.

[CKS09] J. Camenisch, M. Kohlweiss, and C. Soriente. *Public Key Cryptography - PKC 2009: Proc. of the 12th International Conference on Practice and Theory in Public Key Cryptography*, volume 5443 of *Lecture Notes in Computer Science*, chapter An Accumulator Based on Bilinear Maps and Efficient Revocation for Anonymous Credentials, pages 481–500. 2009.

[CL01] J. Camenisch and A. Lysyanskaya. An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation. In *Proc. of the International Conference on the Theory and Application of Cryptographic Techniques: Advances in Cryptology (EUROCRYPT)*, pages 93–118, 2001.

[CLLW08] Y. Cao, Y. Li, H. Li, and X. Wang. An anonymous authentication protocol for privacy protection in location based services. In *Proc. of the 4th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*, pages 1–5, 2008.

[CMGK14] J. Calabuig, J.F. Monserrat, D. Gozalvez, and O. Klemp. Safety on the Roads: LTE Alternatives for Sending ITS Messages. *IEEE Vehicular Technology Magazine*, 9(4):61–70, 2014.

[CON] CONVERGE. Project Website. `http://www.converge-online.de` (accessed in 04.2016).

[CON15a] CONVERGE - COmmunication Network VEhicle Road Global Extension. Architecture of the Car2X Systems Network. Deliverable D4.3, 2015.

[CON15b] CONVERGE - COmmunication Network VEhicle Road Global Extension. Final Operational Requirements and Role Models. Deliverable D1.2, 2015.

[CVH02] J. Camenisch and E. Van Herreweghen. Design and Implementation of the Idemix Anonymous Credential System. In *Proc. of the 9th ACM Conference on Computer and Communications Security (CCS)*, pages 21–30, 2002.

[Dah14] J. Daher. Path Hiding in der C2X-Kommunikation. Master's Thesis, University of Rostock, 2014.

[dB14] P. M. d'Orey and M. Boban. Empirical Evaluation of Cooperative Awareness in Vehicular Communications. In *Proc. of the 79th IEEE Vehicular Technology Conference (VTC Spring)*, pages 1–5, 2014.

[DH98] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460, December 1998.

[dMdlIZ15] P. M. d'Orey, N. Maslekar, I. de la Iglesia, and N. K. Zahariev. Navi: Neighbor-aware virtual infrastructure for information collection and dissemination in vehicular networks. In *Proc. of the 81th IEEE Vehicular Technology Conference (VTC Spring)*, pages 1–6, 2015.

[DRI14] DRIVE C2X - Accelerate cooperative mobility. Final Report (IP-Deliverable). Deliverable D11.6, 2014.

[Dub15] R. Dubitzky. The Car as a Sensor: Cooperative Perception and Learning for Automated Driving. *ATZelektronik worldwide*, 10(2):18–21, 2015.

[EGH+08] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan. The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring. In *Proc. of the 6th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 29–39, 2008.

[ETS09a] ETSI. Intelligent Transport Systems (ITS); European profile standard for the physical and medium access control layer of Intelligent Transport Systems operating in the 5 GHz frequency band. Final draft ETSI ES 202 663 V1.1.0, European Telecommunications Standards Institute, November 2009.

[ETS09b] ETSI. Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Definitions. ETSI TR 102 638 V1.1.1, European Telecommunications Standards Institute, June 2009.

[ETS10a] ETSI. Intelligent Transport Systems (ITS); Facilities Layer Function, Part 2: Services Announcement. Draft ETSI TS 102 890-2 V0.0.3, European Telecommunications Standards Institute, February 2010.

[ETS10b] ETSI. Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service. ETSI TS 102 637-3 V1.1.1, European Telecommunications Standards Institute, September 2010.

[ETS10c] ETSI. Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 1: Requirements. ETSI TS 102 636-1 V1.1.1, European Telecommunications Standards Institute, March 2010.

[ETS11a] ETSI. Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service. ETSI TS 102 637-2 V1.2.1, European Telecommunications Standards Institute, March 2011.

[ETS11b] ETSI. Intelligent Transport Systems (ITS); Vehicular Communications; Geographical Area Definition. ETSI EN 302 931 V1.1.1, European Telecommunications Standards Institute, July 2011.

[ETS11c] ETSI. Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 6: Internet Integration; Sub-part 1: Transmission of IPv6 Packets over GeoNetworking Protocols. ETSI TS 102 636-6-1 V1.1.1, European Telecommunications Standards Institute, March 2011.

[ETS12a] ETSI. Intelligent Transport Systems (ITS); Framework for Public Mobile Networks in Cooperative ITS (C-ITS). ETSI TR 102 962 V1.1.1, European Telecommunications Standards Institute, February 2012.

[ETS12b] ETSI. Intelligent Transport Systems (ITS); Security; Trust and Privacy Management. ETSI TS 102 941 V1.1.1, European Telecommunications Standards Institute, June 2012.

[ETS13] ETSI. Intelligent Transport Systems (ITS); Security; Security header and certificate formats. ETSI TS 103 097 V1.1.1, European Telecommunications Standards Institute, April 2013.

[ETS14a] ETSI. Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 4: Geographical addressing and forwarding for point-to-point and point-to-multipoint communications; Sub-part 1: Media-Independent Functionality. Draft ETSI EN 302 636-4-1 V1.2.1, European Telecommunications Standards Institute, February 2014.

[ETS14b] ETSI. Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 5: Transport protocols; Sub-part 1: Basic Transport Protocol. Draft ETSI EN 302 636-5-1 V1.2.1, European Telecommunications Standards Institute, February 2014.

[ETS15] ETSI. Intelligent Transport Systems (ITS); Security; Security header and certificate formats. ETSI TS 103 097 V1.2.1, European Telecommunications Standards Institute, June 2015.

[ETS16a] ETSI. Mobile Edge Computing (MEC); Framework and Reference Architecture. ETSI GS MEC 003 V1.1.1, European Telecommunications Standards Institute, March 2016.

[ETS16b] ETSI. Mobile Edge Computing (MEC); Technical Requirements. ETSI GS MEC 002 V1.1.1, European Telecommunications Standards Institute, March 2016.

[ETS16c] ETSI. Mobile Edge Computing (MEC); Terminology. ETSI GS MEC 001 V1.1.1, European Telecommunications Standards Institute, March 2016.

[FKL16] D. Förster, F. Kargl, and H. Löhr. PUCA: A pseudonym scheme with strong privacy guarantees for vehicular ad-hoc networks. *Ad Hoc Networks*, 37, Part 1:122 – 132, 2016.

[FLK15] D. Förster, H. Löhr, and F. Kargl. Decentralized enforcement of k-anonymity for location privacy using secret sharing. In *Proc. of the 2015 IEEE Vehicular Networking Conference (VNC)*, pages 279–286, 2015.

[For05] Forschungsgesellschaft für Straßen- und Verkehrswesen. *Handbuch für die Bemessung von Straßenverkehrsanlagen (HBS)*. 2005.

[FPK13] M. Feiri, J. Petit, and F. Kargl. Efficient and Secure Storage of Private Keys for Pseudonymous Vehicular Communication. In *Proc. of the 2013 ACM Workshop on Security, Privacy & Dependability for Cyber Vehicles (CyCAR)*, pages 9–18, 2013.

[FRH09] J. Freudiger, M. Raya, and J.-P. Hubaux. Self-organized Anonymous Authentication in Mobile Ad Hoc Networks. In *Security and Privacy in Communication Networks*, volume 19 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 350–372. 2009.

[Fut] FutureID. Project Website. `http://www.futureid.eu` (accessed in 04.2016).

[FWF13] R. L. Finn, D. Wright, and M. Friedewald. *European Data Protection: Coming of Age*, chapter Seven Types of Privacy, pages 3–32. 2013.

[FWZ12] A. Festag, M. Wiecker, and N. Zahariev. Safety and traffic efficiency applications for geomessaging over cellular mobile networks. In *Proc. of the 19th ITS World Congress*, 2012.

[GM] GM. OnStar Services. `https://www.onstar.com/us/en/services/services.html` (accessed in 04.2016).

[GM 01] GM Heritage Center. 1996, OnStar. `https://history.gmheritagecenter.com/wiki/index.php/1996,_OnStar` (accessed in 04.2016), 2001.

[GM14] GM. Chevrolet AppShop Offers Connected Car Personalization. `http://media.gm.com/media/us/en/gm/news.detail.html/content/Pages/news/us/en/2014/Jan/0105-gm-appshop.html` (accessed in 04.2016), 2014.

[HER16] HERE. HD Live Map. `https://lts.cms.here.com/static-cloud-content/Company_Site/2016_01/160205_HERE_HDLiveMap_USLetterRGB.pdf` (accessed in 04.2016), 2016.

[Hes10] Hessen Mobil. Was ist los auf Hessens Straßen? Straßenverkehrszählung. 2010.

[HG05] Baik Hoh and M. Gruteser. Protecting Location Privacy Through Path Confusion. In *Proc. of the 1st IEEE International Conference on Security and Privacy for Emerging Areas in Communications Networks (SecureComm)*, pages 194–205, 2005.

[HKH10] Kuan Lun Huang, Salil S. Kanhere, and Wen Hu. Preserving privacy in participatory sensing systems. *Computer Communications*, 33(11):1266–1280, 2010.

[HPSK13] C. Höfer, J. Petit, R. Schmidt, and F. Kargl. POPCORN: Privacy-preserving Charging for Emobility. In *Proc. of the 2013 ACM Workshop on Security, Privacy & Dependability for Cyber Vehicles (CyCar)*, pages 37–48, 2013.

[HSL+14] A. Hannak, G. Soeller, D. Lazer, A. Mislove, and C. Wilson. Measuring Price Discrimination and Steering on E-commerce Web Sites. In *Proc. of the 2014 Conference on Internet Measurement Conference (IMC)*, pages 305–318, 2014.

[IEE04] IEEE. IEEE Standard Specifications for Public-Key Cryptography - Amendment 1: Additional Techniques. IEEE Std 1363a-2004 (Amendment to IEEE Std 1363-2000), Institute of Electrical and Electronics Engineers, 2004.

[IEE10] IEEE. 802.11p-2010 - IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments. IEEE Std 802.11p-2010 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, IEEE Std 802.11n-2009, and IEEE Std 802.11w-2009), Institute of Electrical and Electronics Engineers, 2010.

[IEE13a] IEEE. IEEE Guide for Wireless Access in Vehicular Environments (WAVE) - Architecture. IEEE 1609.0-2013, Institute of Electrical and Electronics Engineers, 2013.

[IEE13b] IEEE. Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages. IEEE 1609.2-2013, Institute of Electrical and Electronics Engineers, 2013.

[JH11] A. Jaeger and S. A. Huss. The weather hazard warning in simTD: A design for road weather related warnings in a large scale Car-to-X field operational test. In *Proc. of the 11th International Conference on ITS Telecommunications (ITST)*, pages 375–380, 2011.

[JMV01] D. Johnson, A. Menezes, and S. A. Vanstone. The Elliptic Curve Digital Signature Algorithm (ECDSA). *International Journal of Information Security*, 1(1):36–63, 2001.

[JRX11] G. Jodlauk, R. Rembarz, and Z. Xu. An Optimized Grid-Based Geocasting Method for Cellular Mobile Networks. In *Proc. of the 18th ITS World Congress*, 2011.

[KEBB12] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker. Recent Development and Applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5(3&4):128–138, 2012.

[KJP14] M. Khodaei, H. Jin, and P. Papadimitratos. Towards deploying a scalable amp; robust vehicular identity and credential management infrastructure. In *Proc. of the 2014 IEEE Vehicular Networking Conference (VNC)*, pages 33–40, 2014.

[Kru07] J. Krumm. Inference Attacks on Location Tracks. In *Proc. of the 5th IEEE International Conference on Pervasive Computing (PERVASIVE)*, pages 127–143, 2007.

[KWC14] W.-C. Kuo, H.-J. Wei, and J.-C. Cheng. An efficient and secure anonymous mobility network authentication scheme. *Journal of Information Security and Applications*, 19(1):18–24, 2014.

[Lae15] J. Laenge. Evaluation verschiedener Strategien zur anonymen Veröffentlichung georeferenzierter Sensordaten. Bachelor's Thesis, Hochschule Darmstadt – University of Applied Sciences, 2015.

[LHH08] K. P. Laberteaux, J. J. Haas, and Y.-C. Hu. Security Certificate Revocation List Distribution for Vanet. In *Proc. of the 5th ACM International Workshop on VehiculAr Inter-NETworking (VANET)*, pages 88–89, 2008.

[LLZ+07] X. Lin, R. Lu, H. Zhu, P.-H. Ho, X. Shen, and Z. Cao. ASRPAKE: An Anonymous Secure Routing Protocol with Authenticated Key Exchange for Wireless Ad Hoc Networks. In *Proc. of the 2007 IEEE International Conference on Communications (IEEE ICC)*, pages 1247–1253, 2007.

[Mai04] C. Maihofer. A survey of geocast routing protocols. *IEEE Communications Surveys Tutorials*, 6(2):32–42, 2004.

[Nak08] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. `https://bitcoin.org/bitcoin.pdf` (accessed in 04.2016), 2008.

[NI97] J. C. Navas and T. Imielinski. GeoCast - Gographic Addressing and Routing. In *Proc. of the 3rd annual ACM/IEEE international conference on Mobile computing and networking (MobiCom)*, pages 66–76, 1997.

[NIS] NIST. Specification for the ADVANCED ENCRYPTION STANDARD (AES). Federal information processing standards publication, National Institute of Standards and Technology.

[NL08] D. K. Nilsson and U. E. Larson. Secure Firmware Updates over the Air in Intelligent Vehicles. In *Proc. of the 2008 IEEE International Conference on Communications Workshops*, pages 380–384, 2008.

[OYN+08] H. Oguma, A. Yoshioka, M. Nishikawa, R. Shigetomi, A. Otsuka, and H. Imai. New Attestation Based Security Architecture for In-Vehicle Communication. In *Proc. of the IEEE Global Telecommunications Conference, (GLOBECOM)*, pages 1–6, 2008.

[PHJOZ15] Asier Perallos, Unai Hernandez-Jayo, Enrique Onieva, and Ignacio Julio García Zuazola, editors. *Intelligent Transport Systems: Technologies and Applications*. 2015.

[Pos80] J. Postel. User Datagram Protocol. RFC 768, August 1980.

[Pos81] J. Postel. Transmission Control Protocol. RFC 793, September 1981.

[PRI] PRIME. Project Website. `http://www.prime-project.eu` (accessed in 04.2016).

[PW12] T. Pögel and L. Wolf. Analysis of operational 3G network characteristics for adaptive vehicular Connectivity Maps. In *2012 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pages 355–359, 2012.

[PZ13] C. Paquin and G. Zaverucha. U-Prove Cryptographic Specification V1.1 (Revision 3), 2013.

[QWC13] F. Qiu, F. Wu, and G. Chen. SLICER: A Slicing-Based K-Anonymous Privacy Preserving Scheme for Participatory Sensing. In *Proc. of the 10th International IEEE Conference on Mobile Ad-Hoc and Sensor Systems (MASS)*, pages 113–121, 2013.

[QWC15] F. Qiu, F. Wu, and G. Chen. Privacy and Quality Preserving Multimedia Data Aggregation for Participatory Sensing Systems. *IEEE Transactions on Mobile Computing*, 14(6):1287–1300, 2015.

[RST01] R. L. Rivest, A. Shamir, and Y. Tauman. How to Leak a Secret. In *Proc. of the 7th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT)*, pages 552–565, 2001.

[SAE16a] SAE. Dedicated Short Range Communications (DSRC) Message Set Dictionary™. SAE J2735 2016-03, Society of Automotive Engineers, March 2016.

[SAE16b] SAE. On-Board System Requirements for V2V Safety Communications. SAE J2945/1 2016-03, Society of Automotive Engineers, March 2016.

[Sch11] Björn Schünemann. V2X simulation runtime infrastructure VSimRTI: An assessment tool to design smart traffic management systems. *Computer Networks*, 55(14):3189–3198, 2011.

[sim13] simTD - Sichere Intelligente Mobilität Testfeld Deutschland. TP5-Abschlussbericht. Deliverable D5.5, 2013.

[Sin12] A. Singh. Restricted Usage of Anonymous Credentials in VANET for Misbehavior Detection. Master's thesis, Frankfurt University of Applied Sciences, 2012.

[Swe02] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[SZ11] D. Shang and Q. Zhou. Dynamic mbsfn area configuration method in consideration of radio resource efficiency and system thereof. Patent, US 20110007668, 2011.

[Urm14] C. Urmson. The latest chapter for the self-driving car: mastering city street driving. `https://googleblog.blogspot.de/2014/04/the-latest-chapter-for-self-driving-car.html` (accessed in 04.2016), 2014.

[VDA15] VDA - Verband der Automobilindustrie. Automation - From Driver Assistance Systems to Automated Driving, 2015.

[VMA00] Scott A. Vanstone, Ronald C. Mullin, and Gordon B. Agnew. Elliptic curve encryption systems. Patent, US 6141420, 2000.

[VRBZ08] D. Valerio, F. Ricciato, P. Belanovic, and T. Zemen. UMTS on the Road: Broadcasting Intelligent Road Safety Information via MBMS. In *Proc. of the IEEE Vehicular Technology Conference (VTC Spring)*, pages 3026–3030, 2008.

[WB90] S. D. Warren and L. D. Brandeis. The right to privacy. *Harvard Law Review*, 4(5):193–220, 1890.

[Wes67] A. F. Westin. *Privacy and Freedom*. 1967.

[WK13] L. Wang and G. S. G. S. Kuo. Mathematical modeling for network selection in heterogeneous wireless networks - a tutorial. *IEEE Communications Surveys Tutorials*, 15(1):271–292, 2013.

[WMKP10] B. Wiedersheim, Zhendong Ma, F. Kargl, and P. Papadimitratos. Privacy in inter-vehicular networks: Why simple pseudonym change is not enough. In *Proc. of the 7th International Conference on Wireless On-demand Network Systems and Services (WONS)*, pages 176–183, 2010.

[WWKH13] W. Whyte, A. Weimerskirch, V. Kumar, and T. Hehn. A security credential management system for v2v communications. In *Proc. of the 2013 IEEE Vehicular Networking Conference (VNC)*, pages 1–8, 2013.

[ZBS+14] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, Thao Dang, U. Franke, N. Appenrodt, C. G. Keller, E. Kaus, R. G. Herrtwich, C. Rabe, D. Pfeiffer, F. Lindner, F. Stein, F. Erbs, M. Enzweiler, C. Knöppel, J. Hipp, M. Haueis, M. Trepte, C. Brenk, A. Tamke, M. Ghanaat, M. Braun, A. Joos, H. Fritz, H. Mock, M. Hein, and E. Zeeb. Making Bertha Drive - An Autonomous Journey on a Historic Route. *IEEE Intelligent Transportation Systems Magazine*, 6(2):8–20, 2014.

Curriculum Vitae

Personal Details

Name	Carsten Gerhard Büttner
Date of Birth	2nd April 1987
Place of Birth	Aschaffenburg
Nationality	German

Education

08/2012 - 11/2016	Ph.D. Candidate, TU Darmstadt
11/2011 - 09/2012	IT Security, TU Darmstadt, Master
08/2010 - 05/2011	Exchange Student, Nanyang Technological University, Singapore
04/2010 - 12/2011	Information System Technology, TU Darmstadt, Master
10/2006 - 03/2010	Information System Technology, TU Darmstadt, Bachelor
09/2004 - 07/2006	Fachoberschule Aschaffenburg, Fachabitur
09/2000 - 08/2004	Edith-Stein-Realschule Alzenau, Realschulabschluss
09/1997 - 07/2000	Hauptschule Schöllkrippen
09/1993 - 07/1997	Grundschule Schöllkrippen

Work Experience

since 03/2016	Lead Engineer Product Cybersecurity, Adam Opel AG
08/2012 - 02/2016	Research Engineer, Adam Opel AG
02/2007 - 07/2012	Software Engineer, Spedion GmbH

Professional Services

Program Committee	escar Europe 2016 escar USA 2017

Publications and Talks

Book Chapters

B.2 **Carsten Büttner** and Sorin A. Huss. An Efficient Approach to Anonymous Distribution of ITS Messages in Geographic Areas via LTE and IRS Networks. Olivier Camp, Steven Furnell, and Paolo Mori, editors, Information Systems Security and Privacy: Second International Conference, ICISSP 2016, Rome, Italy, February 19-21, 2016, Revised Selected Papers (Communications in Computer and Information Science), to appear.

B.1 **Carsten Büttner** and Sorin A. Huss. An Efficient Anonymous Authenticated Key Agreement Protocol for Vehicular Ad-Hoc Networks Based on Ring Signatures and the Elliptic Curve Integrated Encryption Scheme. Olivier Camp, Edgar Weippl, Christophe Bidan, and Esma Aïmeur, editors, Information Systems Security and Privacy: First International Conference, ICISSP 2015, Angers, France, February 9-11, 2015, Revised Selected Papers (Volume 576 of Communications in Computer and Information Science), pages 139–159, January 2016.

Full Conference Papers

C.8 **Carsten Büttner** and Sorin A. Huss. Attribute-Based Authorization Tickets for Car-to-X Communication, *2016 IEEE Conference on Communications and Network Security (CNS 2016)*, Philadelphia, USA, October 2016

C.7 **Carsten Büttner**, Tobias Rückelt, and Sorin A. Huss. Sicheres Hochladen, Austauschen und Verteilen von Daten in einem Car2X Systemverbund. *7. VDE GMM-Fachtagung Automotvie meets Electronics (AmE 2016)*, Dortmund, Germany, March 2016.

C.6 **Carsten Büttner** and Sorin A. Huss. An Anonymous Geocast Scheme for ITS Applications. *2nd International Conference on Information Systems Security and Privacy (ICISSP 2016)*, Rome, Italy, February 2016.
→ Best Student Paper Award Nominee

C.5 **Carsten Büttner**, Friederike Bartels, and Sorin A. Huss. Real-World Evaluation of an Anonymous Authenticated Key Agreement Protocol for Vehicular Ad-Hoc Networks. *11th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2015)*, Abu Dhabi, UAE, October 2015.

C.4 **Carsten Büttner** and Sorin A. Huss. Path Hiding for Privacy Enhancement in Vehicular Ad-Hoc Networks. *82nd IEEE Vehicular Technology Conference (VTC2015-Fall)*, Boston, USA, September 2015.

C.3 Tobias Rückelt and **Carsten Büttner**. Secure Service Management for ITS-Services. *6. VDE GMM-Fachtagung Automotvie meets Electronics (AmE 2015)*, Dortmund, Germany, February 2015.

C.2 **Carsten Büttner** and Sorin A. Huss. A Novel Anonymous Authenticated Key Agreement Protocol for Vehicular Ad Hoc Networks. *1st International Conference on Information Systems Security and Privacy (ICISSP 2015)*, Angers, France, February 2015.
→ Best Student Paper Award

C.1 **Carsten Büttner**, Harald Berninger, and Sorin A. Huss. Security und Privacy in CONVERGE am Beispiel eines Kommunikationsszenarios. *5. VDE GMM-Fachtagung Automotvie meets Electronics (AmE 2014)*, Dortmund, Germany, February 2014.

Workshop Papers

W.1 Delphine Christin, **Carsten Büttner**, and Nicolas Repp. CachedSensing: Exploring and documenting the environment as a treasure hunt. *7th IEEE International Workshop on Practical Issues in Building Sensor Network Applications (SenseApp 2012)*, Clearwater, USA, October 2012.

Technical Reports

T.1 **Carsten Büttner** and Sorin A. Huss. Anonymous Credentials and Attribute-Based Authorization Tickets in Car-to-X Communication. *2nd GI/ITG KuVS Fachgespräch Inter-Vehicle Communication (FG-IVC 2014)*, Luxembourg City, Luxembourg, February 2014.

Patents

P.1 **Carsten Büttner** and Sorin A. Huss. Verfahren zum Verbreiten einer Nachricht. DE102015009599.4, Patent Pending, July 2015.

Talks

T.3 **Carsten Büttner**. IT-Security – Continuative Approaches. *CONVERGE: Presentation of Scientific Results*, BMWI Berlin, October 2015.

T.2 **Carsten Büttner**. Intelligent Transportation Systems. *Ambient Intelligence*. TU Darmstadt, February 2015.

T.1 **Carsten Büttner**. Intelligent Transportation Systems. *Ambient Intelligence*. TU Darmstadt, February 2014.

Supervising Activities

Master's Thesis

12/2014 - 05/2015	Yasser Aziza (Technische Universität Darmstadt): Anonymous Credentials in Car-to-X Communication
02/2014 - 08/2014	Josef Daher (University of Rostock): Path Hiding in der C2X-Kommunikation

Bachelor's Thesis

05/2015 - 07/2015	Jakob Laenge (Hochschule Darmstadt – University of Applied Sciences): Evaluation verschiedener Strategien zur anonymen Veröffentlichung georeferenzierter Sensordaten
02/2015 - 09/2015	Friederike Bartels (University of Applied Sciences Wiesbaden Rüsselsheim): Implementation and Evaluation of an Anonymous Authenticated Key Agreement Protocol in Vehicular Ad Hoc Networks
05/2014 - 11/2014	Johannes Wagener (Technische Universität Darmstadt): Implementierung von Ring-Signaturen für die Car2X-Kommunikation unter Verwendung von sicherem Speicher
03/2014 - 08/2014	Christoph Brenner (University of Applied Sciences Wiesbaden Rüsselsheim): Sensordatenübertragung

Internship's

08/2015 - 11/2015	Marc Schiller (Technische Universität Darmstadt): Erweiterung und Optimierung eines Anonymen Key Agreement Protokolls

02/2015 - 04/2015	Jakob Laenge (Hochschule Darmstadt – University of Applied Sciences): Pfadrekonstruktion anhand von anonymisierten geogetaggten Daten
10/2014 - 01/2015	Matthias Braun (University of Applied Sciences Wiesbaden Rüsselsheim): Auswertung und Visualisierung von Sensordaten
08/2014 - 01/2015	Friederike Bartels (University of Applied Sciences Wiesbaden Rüsselsheim): Senden und Empfangen von C2X-GeoMulticast-Nachrichten
09/2013 - 02/2014	Christoph Brenner (University of Applied Sciences Wiesbaden Rüsselsheim): Sensordatenverarbeitung in C
05/2013 - 11/2013	Josef Daher (University of Rostock): IT Sicherheit und Privatsphäre in der Car2X-Kommunikation